

Grafisk fremstilling af fraktaler og kaos

Christiansen, Peder Voetmann

Publication date:
1989

Document Version
Også kaldet Forlagets PDF

Citation for published version (APA):
Christiansen, P. V. (1989). *Grafisk fremstilling af fraktaler og kaos*. Roskilde Universitet. Tekster fra IMFUFA Nr. 174

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

Take down policy

If you believe that this document breaches copyright please contact rucforsk@kb.dk providing details, and we will remove access to the work immediately and investigate your claim.

TEKST NR 174

1989

Grafisk fremstilling af

FRAKTALER OG KAOS.

Peder Voetmann Christiansen.



TEKSTER fra

IMFUFA

ROSKILDE UNIVERSITETSCENTER
INSTITUT FOR STUDIET AF MATEMATIK OG FYSIK SAMT DERES
FUNKTIONER I UNDERVISNING, FORSKNING OG ANVENDELSER

IMFUFA, Roskilde Universitetscenter, Postbox 260, 4000 Roskilde
Grafisk fremstilling af FRAKTALER OG KAOS.
af: Peder Voetmann Christiansen

IMFUFA tekst nr. 174/89

76 sider

ISSN 0106-6242

Abstract .

Teksten introducerer nogle simple eksempler på systemer med kaotisk adfærd og diskuterer, hvordan man grafisk skelner en sådan fra andre typer af bevægelse, f.eks. næstenperiodiske. Desuden gives en række eksempler på fraktaler med eksakt eller tilnærmet selv-similaritet, og den fraktale dimension introduceres ved skala-invarians betragtninger.

Det er meningen, at teksten skal afprøves i gymnasieklasser i forbindelse med det matematisk-datalogiske emne i løbet af foråret 89, og til efteråret eventuelt følges op med en revideret version, udgivet af matematiklærerforeningen.

Forord .

Denne tekst er beregnet til det matematisk-datalogiske emne i 3. gymnasieklasse. Studiet af fraktaler og kaotiske dynamikker indbyder til en eksperimentel tilgang til matematikken, hvor datamaskinen benyttes som en form for mikroskop, der giver indblik i komplicerede strukturer. Der er derfor ikke gjort meget ud af matematisk bevisførelse, men det er mit håb, at den selvstændige produktion af overraskende grafiske udtryk med simple programmer kan give appetit til at gå på matematisk opdagelse.

Jeg har lagt vægt på simple iterative og rekursive metoder, som ikke kræver kendskab til komplekse tal. "Figentræet og mandelbrødet" er et sammenhængende emne i kaos-fraktal-teorien, som behandles i en anden tekst, udgivet af matematiklærerforeningen. Figentræet er derfor et af de oplagte emner, som er udeladt af denne tekst. Der findes dog et figentræ-program på den medfølgende diskette. En anden tekst, Henrik Darlies "fraktaler i Comal", som kan rekvireres via Amtscentralen, ledsages også af en diskette. Heri findes en mængde udmærkede programmer, som vil kunne supplere eksemplerne i denne tekst.

Et hovedemne i nærværende tekst er begrebet "fraktal dimension", som indføres i forbindelse med strengt skalainvariante eller selv-similære figurer, frembragt ved rekursive procedurer. Et 20 timers undervisningsforløb bør bygges op om denne kerne, således at alle gennemfører de rekursive stregtegninger, gennemgået i kapitel 4 (max. 10 timer). Den resterende del af forløbet kan så benyttes til videregående undersøgelse, f.eks. af iterationsformlerne i kap. 1 og 2, som indbyder til kunstneriske frembringelser.

eller til et af emnerne DLA (kap. 3), kurver med areal (kap. 5), refleksion fra cylindre (kap. 6), eller bolden i grøften (kap. 7).

Programeksemplerne er skrevet i Uni-Comal til PCere. Der gives ikke nogen introduktion til Comal-sproget, bortset fra den specielle "skildpadde-grafik", som benyttes til de rekursive stregtegninger i kap. 4. Programeksemplerne findes på to disketter, den ene i PC-UniComal, den anden i RC-Comal sammen med en pakke med skildpaddegrafikken. Disse disketter rummer oversættelser af programmer skrevet i andre sprog, først og fremmest mine BASIC programmer til hjemmecomputeren Amstrad CPC 6128. Oversættelserne af de større programmer er udført af Heine Larsen, som jeg hermed takker for den store hjælp, han har ydet mig.

Figurerne i teksten er frembragt af en almindelig matrix printer, koblet til min hjemmecomputer (bortset fra fig. 14, som er lavet med plotter). Der er benyttet 2 forskellige formater: et groft format med 320*200 punkter inden for billedrammen og et fint med 640*600 punkter. Det grove format tillader til gengæld brug af fire "farver", der på de sort-hvide billeder viser sig som forskellige gråtoner. Da de forskellige maskiner afviger meget fra hinanden i deres grafiksystemer, har jeg måttet afstå fra at give konkrete anvisninger på fremstilling af papirbilleder. I teksten forudsætter jeg en CGA-grafik med 640*200 punkter på skærmen. Billeder med 640*600 punkter må så laves ved sammenstykning af tre skærbilleder, hvis maskinen tillader det.

Indholdsfortegnelse .

1. Simpel iteration: Martins formel.	s. 1
2. Itererede funktionssystemer (IFS).	8
3. Fraktal dimension: DLA-støvfnug og prikfraktaler.	19
4. Rekursive stregtegninger: Skildpadde-generatorer.	30
5. Dragekurven: Simple talforhold og puslespil.	43
6. Reflektion fra cylindre.	51
7. Bold i grøft.	56
Appendix A: Programliste, reflektion fra cylindre.	62
Appendix B: Programliste, bold i grøft.	68



Kapitel 1 :

Simpel iteration: Martins formel .

Ordet "iteration" betyder "gentagelse". Iterationsformler har vigtige anvendelser i matematikken, f.eks. til numerisk løsning af ligninger: Man starter med at give et første bud på løsningen. Dette tal indsættes i en iterationsformel, som så giver et nyt (og forhåbentlig bedre) bud, som så igen indsættes, osv. I heldige tilfælde vil gentagen anvendelse af formlen producere tal, som hurtigt konvergerer mod den rigtige løsning, men man kan sagtens komme ud for overraskelser, f.eks. konvergens mod en forkert løsning, eller mangel på konvergens.

Vi skal her se på iterationsformler, som virker på talpar, i stedet for enkelte tal. Et talpar (x,y) kan opfattes som koordinater i en plan, og med passende valg af skala kan det afsættes som et punkt på computerens skærm. En uendelig følge af talpar frembringes ved opskriften

$$\begin{aligned}x_{n+1} &= F(x_n, y_n) \\ y_{n+1} &= G(x_n, y_n)\end{aligned}\tag{1}$$

Følgen må startes med et "frø" (x_0, y_0) . Når man så efter hver anvendelse af (1) afsætter det fremkomne, nye talpar som et punkt på skærmen (forudsat, at det, passende skaleret, falder inden for skærmens rammer), vil der efterhånden fremkomme et mønster af prikker, som i mange tilfælde har en ganske overraskende skønhed og detaljerigdom.

Den vigtigste forudsætning for, at den simple iteration (1) kan føre til interessante billeder, er, at mindst én af

fuunktionerne F og G er ikke-lineær , dvs. involverer andre beregninger end addition og multiplikation med en konstant. Som eksempel ser vi på Martins formel , som først blev publiceret i tidsskriftet Scientific American fra september 1986 (A.K.Dewdney i spalten "Computer recreations"):

$$x_{n+1} = y_n - \text{SGN}(x_n) * \text{SQR}(\text{ABS}(b * x_n - c)) \quad (2)$$

$$y_{n+1} = a - x_n$$

Funktionen G er i dette tilfælde lineær, medens F er ikke-lineær. Der er her benyttet et formelsprog, som minder om computerens Comal, idet multiplikation angives med tegnet *; SGN er fortegnsfunktionen, som er +1, 0 eller -1, når argumentet er hhv. positivt, nul, eller negativt; SQR er kvadratrodsfunktionen, og ABS er absolutværdien (den numeriske værdi). Størrelserne a, b og c er såkaldte parametre, som skal holdes konstante under iterationen, men som man bagefter kan forsøge at give andre værdier.

Vi kan benytte nedenstående procedure:

```
PROC martin(a,b,c,x0,y0,n,dx,dy,mx,my)
  x:=x0
  y:=y0
  FOR k:=1 TO n DO
    x1:=y-SGN(x)*SQR(ABS(b*x-c))
    y:=a-x
    x:=x1
    IF ABS(x-mx)<dx AND ABS(y-my)<dy THEN
      PLOT(320*(1+(x-mx)/dx),100*(1+(y-my)/dy))
    ENDIF
  ENDFOR k
ENDPROC martin
```

Her er a , b og c de tre parametre i Martins formel, x_0 og y_0 er startværdierne (frøet), n er antallet af iterationer, og dx , dy , mx og my er konstanter, som angiver billedområdet som et rektangel med midtpunkt i $(x,y)=(mx,my)$ og kantlængderne $2*dx$ og $2*dy$.

Inden proceduren benyttes, skal man fremkalde grafikpakken med kommandoen `USE graphics`. Eventuelt kan grafikmoden med $640*200$ punkter specificeres med kommandoen `graphicscreen(0)`. Prøv så at køre 1000 iterationer med start i $(0,0)$ midt på skærmen. Benyt kommandoen

```
martin(0.1,0.2,-0.3,0,0,1000,10,10,0,0)
```

På fig.1 vises resultatet af en række kørsler med de samme parameterværdier, altså $a=0.1$, $b=0.2$, $c=-0.3$. For hver ny kørsel er startværdierne for x og y tilfældigt valgt.

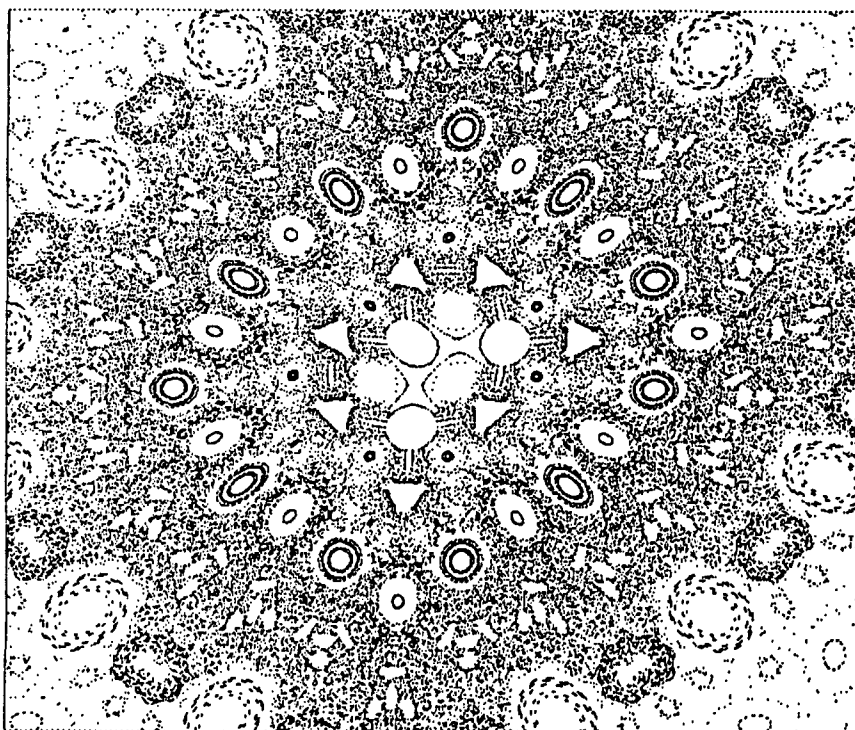


Fig. 1 . Martin: $a=0.1$, $b=0.2$, $c=-0.3$, $dx=5$, $dy=5$, $mx=my=0$.

Når man iagttager skærmen, medens figuren dannes, bemærker man, at hver ny startværdi fører til et bælte af punkter rundt om et centrum. I mange tilfælde vil punkterne danne et system af ellipser (eller ellipselignende lukkede kurver); denne bevægelsesform kaldes næstenperiodisk. Hvis man kunne være så heldig at ramme et af ellipsecentrerne med startværdierne for x og y , ville man få en periodisk bevægelse, som kun afsætter et bestemt antal punkter, ligegyldigt hvor længe, iterationen kører. Når man rammer lidt ved siden af et af disse centre, vil man efter forløbet af perioden ikke ramme startpunktet igen, men lidt ved siden af, og i det lange løb fås aftegnet et antal ellipser, svarende til perioden for den rent periodiske bevægelse. På fig. 1 ses f.eks. to ellipsesystemer med perioden 11 og nær centrum to andre systemer med perioden 3. Ind imellem ellipsesystemerne er der en diffus sværtning med punkter, som indgår i en kaotisk bevægelse.

På fig. 2 er benyttet samme parametre og samme centrum, men værdierne af dx og dy er sat til 1, således at vi får et udsnit omkring midten af fig. 1.

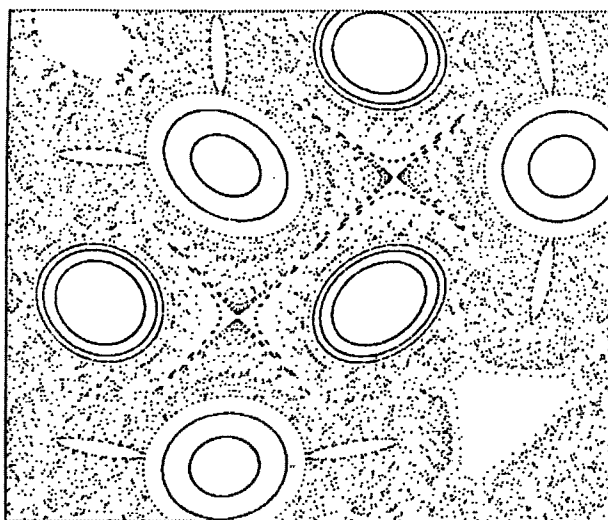


Fig. 2 . Martin: som fig. 1, men $dx=dy=1$.

Vi ser på fig. 2 de to ellipsesystemer med perioden 3, samt en kaotisk fordeling med huller i. Desuden ses to skarpt aftegnede hyperbelsystemer mellem ellipserne. Centrerne af disse hyperbler udgør en ustabil periodisk bevægelse. Når vi rammer lidt ved siden af et af disse centre, vil punktet under sin bevægelse glide langs hyperbelgrenene, og selv om det muligvis i begyndelsen tiltrækkes af centret, vil det uvægerligt senere frastødes og ende i det kaotiske område.

Martins formel er et eksempel på en iteration, der, i modsætning til formlerne for numerisk løsning af ligninger (f.eks. Newton-Raphson), næsten aldrig "falder til ro" dvs. konvergerer mod et enkelt punkt eller et endeligt system af punkter (en såkaldt attraktor). Den minder derved om såkaldt konservative mekaniske systemer, hvor der ingen gnidning er, og hvor den mekaniske energi derfor er bevaret. Vi skal senere, i kapitel 7, se på et sådant system, nemlig en bold, der hopper i en grøft, og i dette eksempel vil vi kunne genfinde mange træk fra Martins iteration. Martins formel, som ikke er tænkt som en model for noget som helst, men snarere som en metode til frembringelse af smukke og gyselige mønstre ("wallpaper for the mind", som Dewdney kalder sin artikel i Scientific American, der også rummer andre eksempler til iteration) rummer imidlertid en utrolig varietet i sit tredimensionale parameterrum, og hvis man går på opdagelse her, kan man være næsten sikker på at frembringe noget, som ingen før har set.

Lad os prøve med nogle andre parameterværdier (foreslået i Dewdneys artikel), nemlig $a=-200$, $b=0.1$ og $c=-80$. I dette tilfælde vil figuren også have et centrum, som imidlertid ligger langt væk fra $(0,0)$. Vi kan bestemme dette centrum som et stationært punkt, dvs. vi indsætter $x_{n+1} = x_n$ og $y_{n+1} = y_n$ i (2) og løser det fremkomne

ligningssystem. Herved bestemmes centret på fig. 3:

$$mx = -95.5547; my = -104.4453 \quad (3)$$

Denne figur viser en mangfoldighed af næstenperiodiske bevægelser: Et enkeltperiodisk ellipsesystem omkring centret (det stationære punkt), som går brat over i et næsten kvadratisk system, som gradvist i større afstande bliver ottekantet. På fig. 4 vises et nærbillede af det underlige overgangsområde fra det cirkulære til det kvadratiske til det ottekantede mønster. I dette område findes såkaldte bifurkationer, dvs. den cirkulære bevægelse omkring centrum går i stykker til et system af usammenhængende cirkelbuer (ordet "cirkel" skal ikke tages for bogstaveligt). Ved langvarige iterationer kan der ophobes numeriske fejl, som på fig. 4 bl.a. viser sig ved mystiske overlapninger mellem de forskellige kurvesystemer.

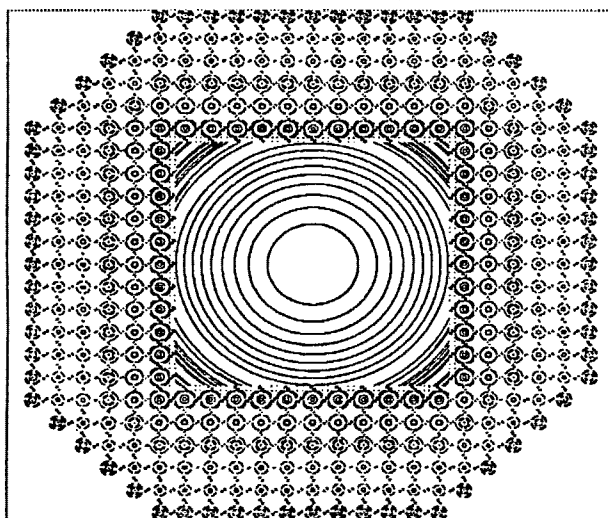


Fig. 3 . Martin: $a=-200$, $b=0.01$, $c=-80$, $dx=213$, $dy=200$, mx og my i det stationære punkt, ligning (3).

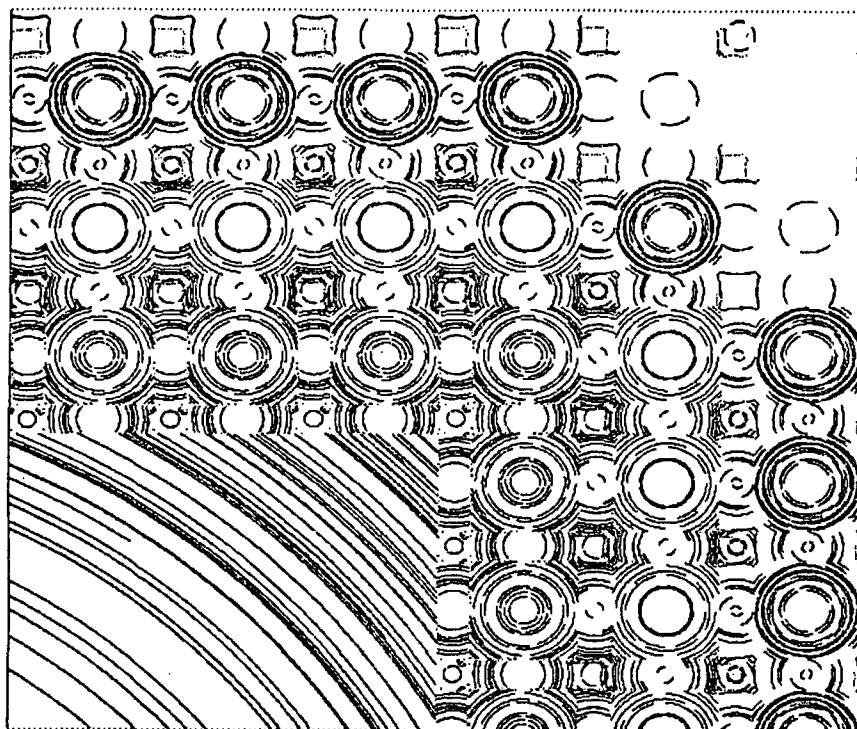


Fig. 4 . Martin: som fig. 3, men centrum (mx, my) i $(0,0)$ og $dx=53$, $dy=50$.

Kapitel 2 :

Itererede funktionssystemer (IFS) .

I stedet for, som i foregående kapitel, at lave en simpel iteration af et enkelt funktionspar (lign. (1)), kan man benytte et system af K funktionspar. For hvert enkelt skridt vælges en tilfældig af funktionerne i systemet:

$$x_{n+1} = F_i(x_n, y_n) \quad (4)$$

$$y_{n+1} = G_i(x_n, y_n)$$

hvor indexet i kan antage værdier mellem 1 og K og skal udvælges tilfældigt for skridt af iterationen. Ud over de parametre, som kan indgå i funktionssættet (F_i, G_i) , skal man så benytte et sæt sandsynligheder til bestemmelse af de relative hyppigheder, hvormed numrene på de respektive funktionspar udvælges. Disse sandsynligheder, $p(i)$ må alle være positive tal mellem 0 og 1, og de skal tilfredsstille betingelsen, at summen af de K sandsynligheder skal være 1. Lad os antage, at vi har givet disse sandsynligheder, $p(i)$ for i fra 1 til K , således at denne betingelse er opfyldt, og lad os overveje, hvordan vi kan få maskinen til at foretage et tilfældigt valg, svarende til disse sandsynligheder.

I Comal er der en funktion RND, som frembringer et tilfældigt tal med jævn sandsynlighedsfordeling mellem 0 og 1. Vi behøver derfor blot at dele intervallet mellem 0 og 1 op i K delintervaller med længder svarende til sandsynlighederne $p(i)$, så kan vi hver gang lade valget afgøres ved nummeret på det interval, som det frembragte tilfældige tal falder indenfor. Hvis vi anbringer

intervallerne i nummerorden, således at det første interval har venstre endepunkt i 0 og højre endepunkt i $p(1)$, kan vi bestemme de højre endepunkter $cp(i)$ for intervallerne 1-K ved programmet

```
cp(1):=p(1)
FOR i:=2 TO K DO cp(i):=cp(i-1)+p(i)
```

Det sidste interval har så højre endepunkt $cp(K)=1$. Et tilfældigt valg af et i mellem 1 og K i overensstemmelse med sandsynlighederne $p(i)$ kan så foretages på følgende måde:

```
z:=RND
i:=1
WHILE cp(i)<z DO i:=i+1
```

Hvis alle sandsynlighederne har samme værdi ($1/K$), er det ikke nødvendigt at udregne størrelserne $cp(i)$, idet valget af i kan foretages ved den simple opskrift

```
i:=1+INT(K*RND)
```

hvor vi har benyttet funktionen $INT(x)$, hvis værdi er det største hele tal, som er mindre end eller lig med x .

I kapitel 1 blev det bemærket, at simpel iteration af en enkelt funktion kræver, at funktionen er ikke-lineær, hvis der skal komme noget ikke-trivielt ud af det. Indførslen af det tilfældige valg mellem flere funktioner i IFS-metoden gør, at selv lineære funktioner kan frembringe interessante, såkaldte fraktale, figurer. De følgende eksempler på IFS benytter kun lineære funktioner, og simpelheden af funktionsudtrykkene giver sig udtryk i en simpel form for lighedannethed mellem del og helhed af

figurerne, som kaldes selv-similaritet. En ret linje er selv-similær, men som påpeget af matematikeren B. Mandelbrot er der mange andre muligheder for selv-similære figurer, og det er i første omgang sådanne figurer, som Mandelbrot indførte betegnelsen fraktaler for.

Vi starter med konstruktion af Serpinski-trekanten. Der er givet tre punkter, $(x_p(1), y_p(1))$, $(x_p(2), y_p(2))$ og $(x_p(3), y_p(3))$, som ikke ligger på samme rette linje. I den på fig. 5 viste Serpinski trekant, udgør disse punkter vinkelspidserne af den store, ligesidede trekant, men det er uvæsentligt, at trekanten er ligesidet. Der skal bruges tre lineære afbildninger af punkterne (x, y) i planen. Afbildning nr. i multiplicerer simpelthen punktets afstande i x - og y -retningen til det faste punkt $(x_p(i), y_p(i))$ med faktoren $\frac{1}{2}$. En sådan afbildning kaldes en affinitet til et punkt. Den enkelte afbildning, samt valget mellem de tre affiniteter (samme sandsynlighed) kan foretages med proceduren:

```
PROC serpin3
  n:=1+INT(3*RND)
  x:=0.5*(x+xp(n))
  y:=0.5*(y+yp(n))
ENDPROC serpin3
```

Denne procedure benyttes i nedenstående program, som starter med at erklære variabelsættene $x_p(i)$ og $y_p(i)$ og tildele dem værdier. Derefter vælges startværdi for (x, y) i trekantens tyngdepunkt midt på skærmen. Der udføres så en såkaldt indsvingning på 1000 iterationer, som foregår "bag kulisserne", dvs. uden at de resulterende punkter afmærkes på skærmen. Først efter, at indsvingningen har elimineret de "forbigående" (transiente) punkter, starter den egentlige iteration, som fortsættes, indtil brugeren trykker på en

tast. Efter afbrydelse af tegningen, vil et nyt tryk på en tast fjerne billedet og bringe tekstskærmen tilbage.

```
DIM xp(3),yp(3)
xp(1):=100
yp(1):=0
xp(2):=540
yp(2):=0
xp(3):=320
yp(3):=200
USE graphics
graphicscreen(0)
x:=320
y:=200
FOR k:=1 TO 1000 DO serpin3
WHILE KEY$<>"" DO NULL
WHILE KEY$="" DO
    serpin3
    plot(x,y)
ENDWHILE
WHILE KEY$<>"" DO NULL
WHILE KEY$="" DO NULL
textscreen
END
```

(Et lille fif: bemærk, at WHILE sætningerne optræder i par, således at karakterbufferen først tømmes for eventuelle indestående karakterer, inden den næste WHILE sætning går i gang med at gøre noget, så længe der ikke trykkes på en tast. I RC-COMAL, skal man i stedet for den tomme streng "" benytte udtrykket CHR\$(0)).

Selv-similariteten af den ligesidede Serpinski-trekant på fig. 5 kan beskrives således: Hvis vi fra trekantens

øverste punkt går ned til midtpunkterne af de omsluttende sider, får vi her en lille trekant, som er ligedannet med den store. Inden i den lille trekant kan vi finde samme forhold. Hvis vi tænker på de afmærkede punkter som havende masse og de hvide områder som masseløse, vil det gælde, at en halvering af sidelængden for en trekant med toppunkt i det øverste punkt af figuren fører til en mindre trekant, hvis masse er en tredjedel af den oprindelige trekants. Eller, hvis vi går fra det mikroskopiske til det makroskopiske: når længdeskalaen forøges med en faktor 2, forøges massen med en faktor 3. Som vi skal diskutere i de følgende kapitler, betyder dette, at vi kan tilskrive Serpinski-trekanten den fraktale dimension (ligedannethedsdimensionen) $\log(3)/\log(2) = 1.585$.

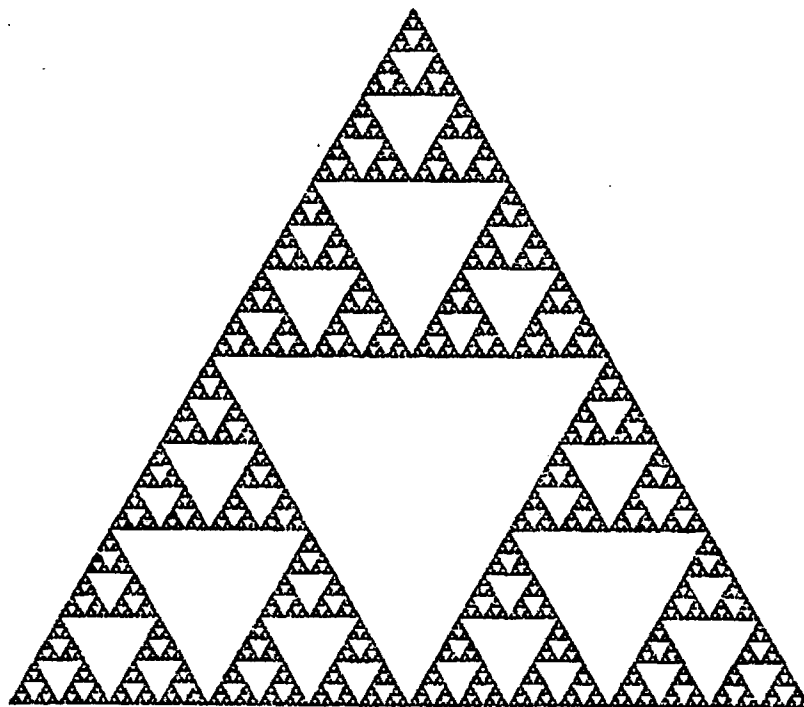


Fig. 5 . Serpinski-trekanten.

Vi kan forestille os, at vi fremstiller en materiel Serpinski-trekant på følgende måde: Vi starter med en ligesidet trekant af krydsfinér. Så forbinder vi midtpunkterne af trekantens sider med linjer og saver langs disse med en løvsav, således at vi får udskåret en trekant i midten. Dernæst gennemfører vi samme operation på de tre tilbageværende trekanter, og igen på de 3×3 resterende trekanter, og således bliver vi ved i en uendelighed.

Lad os antage, at den store trekant, vi startede med, vejede 1 kg. Spørgsmålet er nu: hvor meget vejer alle de udsavede småtrekanter tilsammen? Det kan let udregnes ved simpel matematik. Den første trekant, vi savede ud, har et areal på $1/4$ af den oprindelige trekants, så den må veje $1/4$ kg. De tilbageblevne tre trekanter vejer hver $1/4$ kg, og fra dem skal vi i næste omgang udsave de tre midtertrekanter, som så hver især må veje $(1/4) \cdot (1/4)$ kg. Vi kan da let indse, at den samlede vægt af de uendelig mange trekanter, vi har udsavet, når vi er færdige, må kunne skrives på formen:

$$\text{vægt i kg} = (1/4) \cdot (1 + 3/4 + (3/4)^2 + (3/4)^3 + \dots).$$

Den uendelige sum i parantesen er en såkaldt kvotientrække af typen $1 + q + q^2 + q^3 + \dots$, hvor kvotienten q har værdien $3/4$, dvs. den er numerisk mindre end 1. For en sådan række gælder det, at den uendelige sum har værdien $1/(1-q)$, så vi finder:

$$\text{vægt i kg} = (1/4) \cdot (1/(1 - 3/4)) = 1.$$

De udsavede småtrekanter vejer tilsammen præcis lige så meget som den store trekant, vi startede med! Den resterende Serpinski-trekant, som stadig indeholder en utællelig mangfoldighed af punkter, er en "uendelig tynd" mængde, en såkaldt Cantor-mængde, og det hænger sammen med, at dens fraktale dimension, som vi udregnede til 1.585, er mindre end dimensionen af det rum, den er indlejret i (2).

Det næste eksempel på IFS involverer kun to lineære funktioner med samme sandsynlighed. Resultatet bliver et todimensionalt område, "drageøen", med en indviklet, fraktal "kystlinje". Vi kan benytte følgende procedure:

```
PROC drageifs(x0,y0,scx,scy,n)
  x:=x0
  y:=y0
  FOR k:=1 TO n DO
    x1:=0.5*(x-y)+SGN(RND-0.5)
    y:=0.5*(x+y)
    x:=x1
    plot(scx*x,scy*y)
  ENDFOR k
ENDPROC drageifs
```

Her er (x_0, y_0) startpunktet for punktfølgen (x, y) . Proceduren laver n iterationer og plotter punkterne med brug af skalakonstanterne scx og scy . Det er i dette tilfælde ikke nødvendigt at foretage indsvingning; man skal blot sørge for, at (x_0, y_0) ligger i det indre af øen, som efterhånden bliver helt udfyldt med punkter; dvs. x_0 skal helst ligge mellem $-5/3$ og $5/3$, og y_0 mellem $-1/3$ og $1/3$. Et passende program til tegning af drageøen kan være:

```
USE graphics
graphicscreen(0)
drageifs(0,0,100,50,10000)
```

Med 10000 punkter, som her angivet, får man ikke øen helt udfyldt; der vil være punkter, som først rammes efter en meget langstrakt proces, om nogensinde. En vis tålmodighed kræves, når man laver fraktaler, men alt med måde!

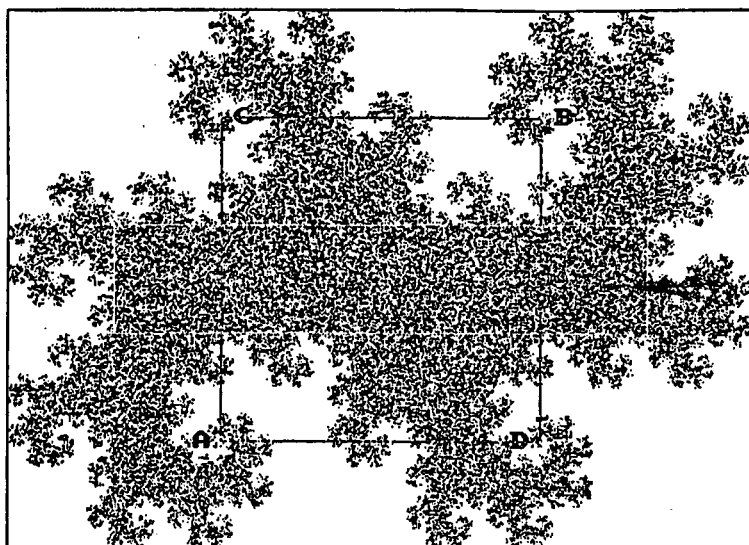


Fig. 6 . Drageøen.

Drageøen har et symmetricentrum i $(0,0)$. På figuren vises tre forskellige rektangler med samme centrum: Det omskrevne rektangel har kantlængderne $14/3$ og $10/3$, og det indskrevne (tyndt hvidt streg) har kantlængderne $10/3$ og $2/3$. Endelig vises et kvadrat med kantlængden 2 og vinkelspidser i punkterne A, C, B og D. Vi skal senere, under omtalen af dragekurven, komme ind på betydningen af disse punkter.

Drageøens kystlinje viser sin fraktale selv-similaritet ved, at store halvøer er besat med små halvøer af samme form, og inden i de store bugter er der små bugter med samme form som de store bugter. Hvad mere er: halvøer og bugter, figur og baggrund udviser de samme formenter. Man kan dække planen med identiske puslespilsbrikker af form som drageøen. Afstanden mellem nabobrikkers centrer må så være $4 (= 7/3 + 5/3)$ i x-retningen og $2 (= 5/3 + 1/3)$ i y-retningen. Vi vender tilbage til problemet om "tiling", dvs. "flisedækning", i forbindelse med dragekurven i kapitel 5.

Som det sidste eksempel på IFS metoden ser vi på konstruktionen af bregnebladet på bogens forside. Til denne, meget naturligt udseende fraktal kræves fire lineære funktionspar (sml. lign. (4)):

$$F_i(x,y) = a_i*x + b_i*y \quad (5)$$

$$G_i(x,y) = c_i*x + d_i*y + e_i$$

hvor i altså kan antage værdierne fra 1 til 4. Der benyttes således 20 parametre $a_1 - - e_4$, men vi kan straks fra starten fastsætte $a_1=b_1=c_1=e_1=0$. Det første funktionspar bliver således en projektion ind på y-aksen, som frembringer den nederste, lodrette del af bregnebladets midterribbe. De resterende 16 parametre kan varieres inden for ret vide rammer, hvorved der kan laves blade med mange forskellige størrelser og krumninger. Det andet og tredje funktionspar har betydning for krumningerne og det fjerde for størrelsen, men vi skal ikke her gå i matematiske detaljer. Parametrene, som er benyttet til tegningen på forsiden, er angivet i nedenstående skema:

i	a	b	c	d	e
1	0	0	0	0.25	0
2	0.85	0.04	-0.04	0.85	1.6
3	0.2	-0.26	0.26	0.22	0.8
4	-0.15	0.28	0.26	0.24	1

Sandsynlighederne for valg af de fire funktionspar betyder ikke noget for bladets endelige form, men er afgørende for fordelingen af punkter i de forskellige områder. Benyt f.eks. $p_1=0.02$, $p_2=0.8$, $p_3=0.08$, $p_4=0.1$.

Den nedenfor angivne procedure forudsætter, at parametrene og de "cumulative" sandsynligheder $cp(i)$ (sml. indledningen til dette kapitel) i forvejen er dimensionerede og tildelt værdier. Ligesom proceduren `serpin3` bestemmer den valget af funktionsparret og opererer en enkelt gang på det givne talpar (x,y) :

```
PROC bladifs
  z:=RND
  i:=1
  WHILE cp(i)<z DO i:=i+1
  x1:=a(i)*x+b(i)*y
  y:=c(i)*x+d(i)*y+e(i)
  x:=x1
ENDPROC bladifs
```

Med de opgivne parameterværdier vil bladet dannes inden for en ramme med x mellem -3 og 3 og y mellem 0 og 12 . Følgende lille program vælger først et tilfældigt startpunkt inden for denne ramme. Dernæst foretages indsvingning, og til sidst plottes (indtil man afbryder ved tryk på en tast) punkterne (x,y) passende skaleret til en skærm med 640×200 punkter. For at udnytte skærmens opløsningsevne bedst er y -værdierne afsat vandret og x -værdierne lodret:

```
x:=6*(RND-0.5)
y:=12*RND
FOR k:=1 TO 100 DO bladifs
  WHILE KEY$<>"" DO NULL
  WHILE KEY$="" DO
    bladifs
    plot(53*y,33*(x+3))
  ENDWHILE
```


Bregnebladet er blot ét eksempel på fraktalgeometriens evne til frembringelse af naturligt udseende objekter. IFS metoden er en af de vigtigste teknikker til dette formål, men der er udviklet andre metoder, som er gode til f.eks. skyer og bjerge. Et fraktalprogram, der kan fremstille vellignende landskabsbilleder kræver et sæt parametre, der specificerer f.eks. bjergkædernes "forrevethed" og skyernes mængde og art (cirrus, cumulus, nimbus). I forhold til det frembragte billedes informationsmængde (128000 bits for et normalt skærbillede med 640*200 punkter) er informationsindholdet i parameterværdierne meget lille (ca 100 bits for bregnebladets parametre). Et fraktalprogram frembringer ny information, eller man skal måske snarere sige, at computeren udfolder information, som i infoldet form, eller "kimform" befinder sig i programmet og parametrene.

Bortset fra de kommercielle anvendelser af sådanne metoder, er der interessante filosofiske og naturvidenskabelige perspektiver i den tanke, at formdannende processer i naturen, f.eks. fosterets udvikling, foregår efter lignende principper. Selve det faktum, at vi med simple fraktalprogrammer og et fåtal af parametre kan lave naturligt udseende billeder, er selvfølgelig ikke noget bevis på, at naturen fungerer på samme måde; det er klart, at virkeligheden er langt mere kompliceret end ethvert program. Men hvis man vil forstå principperne bag naturens dannelse af former, er det nok en god begyndelse, at man i en vis, begrænset forstand kan efterligne dem.

Kapitel 3 :

Fraktal dimension: DLA-støvfnug og prikfraktaler .

I kapitel 1 så vi på en rent deterministisk iterationsproces, dvs. hele forløbet var entydigt bestemt ud fra begyndelsesværdierne af x og y . Med IFS metoden i kapitel 2 introduceredes et element af tilfældighed i valget af et funktionspar for hvert skridt i iterationen. Vi skal nu gå et skridt videre i denne retning og se på en helt indeterministisk proces, hvor tilfældigheden dominerer totalt. Det er den såkaldte DLA-proces, der kan opfattes som en model for dannelsen af støvfnug. Navnet DLA betyder diffusion-limited aggregation .

En samling partikler, der uafhængigt af hinanden udfører tilfældige bevægelser (brownske bevægelser) siges at diffundere. Hvis en sådan samling af partikler til at begynde med er koncentreret i et lille område, vil de under diffusionen efterhånden fjerne sig fra hinanden, ikke fordi de frastøder hinanden (deres bevægelser antages at være uafhængige), men simpelthen fordi en konfiguration, hvor partiklerne ligger tæt sammen, er meget usandsynlig i forhold til en konfiguration, hvor de er mere jævnt spredt ud. Hvis vi anbringer en koncentreret klat farvet væske i et stort rumfang ufarvet væske, vil diffusionen efterhånden udbrede farven, således at klattens areal stort set vokser proportionalt med den forløbne tid.

Vi kan nemt simulere en brownsk bevægelse af en partikel på computerskærmens kvadratiske punktgitter. Vi skal så blot sørge for, at partiklen ikke diffunderer udenfor skærmen. Følgende lille program vælger for hvert skridt en af fire tilfældige bevægelsesretninger, undtagen når partiklen er på randen af skærmen:

```
x:=640*RND
y:=200*RND
plot(x,y)
WHILE KEY$<>"" DO NULL
WHILE KEY$="" DO
    kurs:=INT(4*RND)
    IF kurs=0 AND x<639 THEN x:=x+1
    IF kurs=1 AND y<199 THEN y:=y+1
    IF kurs=2 AND x>0 THEN x:=x-1
    IF kurs=3 AND y>0 THEN y:=y-1
    plot(x,y)
ENDWHILE
```

Da alle de afmærkede punkter bliver stående på skærmen, får man med dette program aftegnet hele banen for den diffunderende partikel. Hvis man kun vil se selve partiklen, må man sørge for at viske det gamle punkt ud, før man plotter det nye.

DLA processen startes ved, at en kim-partikel sættes fast på skærmens midte. Herefter startes en diffunderende partikel fra et tilfældigt sted på randen af skærmen. Denne partikel fortsætter sin bevægelse, indtil den rammer et af de fire nabopunkter til kimpartiklen, hvor den sætter sig fast. Kimen er så vokset til to partikler med ialt 6 nabopunkter. En ny partikel sættes i gang fra skæmranden og diffunderer, indtil den rammer et af disse nabopunkter, hvor den sætter sig fast. Således fortsættes, og efterhånden vokser kimen til en stor, uregelmæssig forgrenet struktur. Denne struktur fortsætter med at vokse, efterhånden som nye partikler sætter sig fast. Væksten sker næsten udelukkende fra spidserne af grenene; det er meget usandsynligt, at en partikel trænger dybt ind i en "fjord", dvs. et mellemrum mellem grene. Således kan vi forestille os, et støvfug

vokser. Når det er blevet stort, vil det have en meget lille massefylde, fordi mellemrummende bliver større, jo længere vi kommer bort fra centret. På fig. 7 vises resultatet af et eksperiment, hvor DLA fnugget er opbygget af 760 partikler.

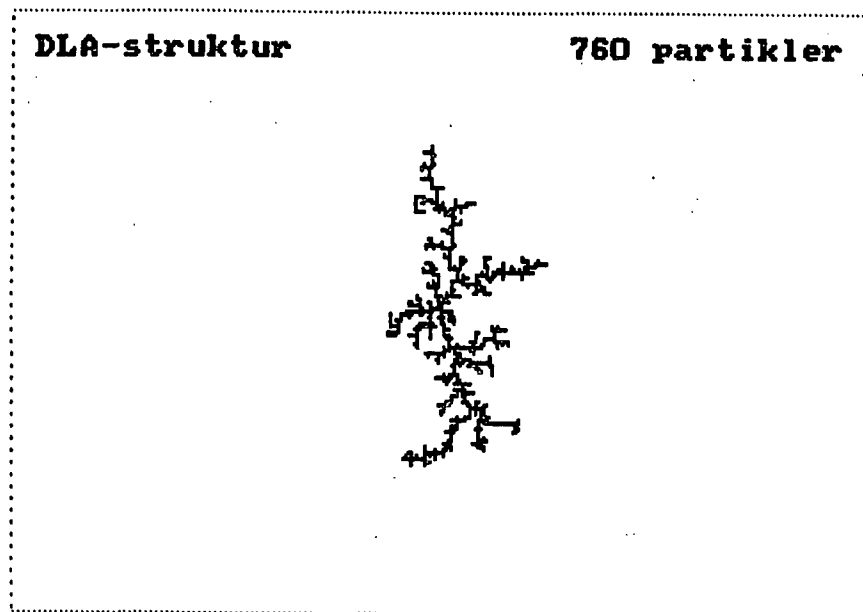


Fig. 7 . DLA-struktur.

For at undersøge fordelingen af partikler i en DLA-struktur, går vi frem på følgende måde. Med centrum i den oprindelige kimpartikel tænker vi os lagt et kvadrat med kantlængden L skærmpunkter, hvor L er et ulige heltal. Vi tæller så, hvor mange partikler, $N(L)$, der befinder sig inden i og på randen af dette kvadrat, og dette gør vi for alle værdier af L fra 1 og op til en så stor værdi, at hele støvfnugget kan være inden i kvadratet. For $L=1$ er $N=1$, idet kimpartiklen jo er i centrum, og når L er så stor, at hele støvfnugget er i kvadratet, er $N(L)$ lig med det totale antal partikler i strukturen. Funktionen $N(L)$ for strukturen på fig. 7 er vist i et dobbeltlogaritmisk diagram på fig. 8.

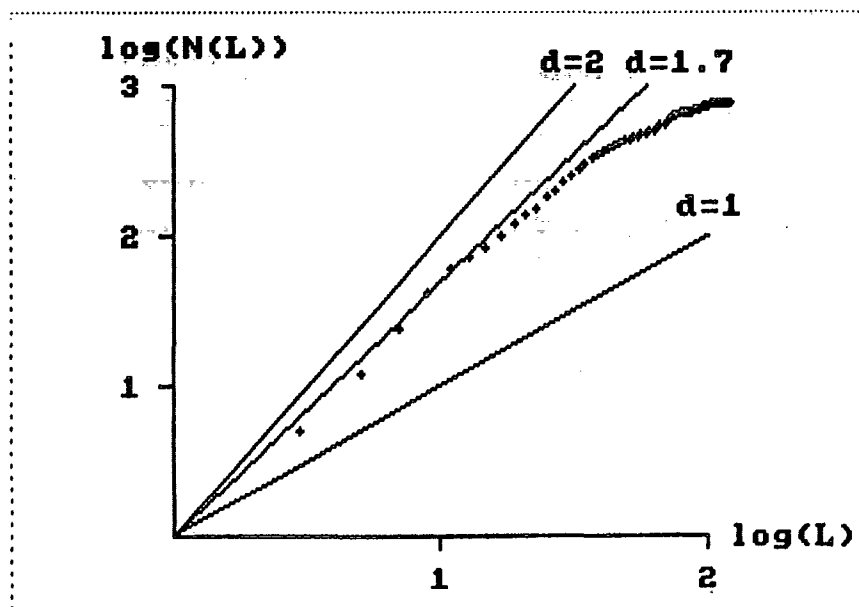


Fig. 8 . Dobbellogaritmisk diagram for $N(L)$ fra fig. 7.

Hvis nu punkterne stort set var jævnt fordelt i planen, ville $N(L)$ vokse proportionalt med arealet af kvadratet, dvs. proportionalt med L^2 , og i det dobbeltlogaritmiske diagram ville punkterne følge en linje med hældningskoefficienten $d=2$. Hvis derimod punkterne hang sammen i en endimensional tråd, ville vi få en linje med hældningskoefficienten $d=1$. Begge disse linjer er indtegnede på fig. 8, og som man kan se, følger punkterne ingen af dem. Derimod følger de i et stort interval ret godt en linje med hældningskoefficienten $d=1.7$. Vi tilskriver derfor DLA-strukturen den fraktale dimension (ligedannethedsdimensionen) 1.7. Forsøg med langt større DLA-fnug (100000 partikler) viser en ret lille spredning af den således bestemte dimension, selv om fnuggene kan falde meget forskellige ud. Strukturen af det underliggende gitter, om det er kvadratisk eller triangulært er underordnet, og det

gør heller ikke noget, om man definerer "naboer" anderledes, således at der er 8 naboer til et punkt på et kvadratisk gitter. Den fraktale dimension af DLA-strukturer afhænger imidlertid af dimensionen af det euklidiske rum, de er indlejrede i: i tre dimensioner fås $d=2.5$.

Lad os overveje, hvad det betyder, når funktionen $N(L)$ i et dobbeltlogaritmisk diagram falder på en ret linje med hældningskoefficienten d . Dvs. der er tale om en potensfunktion

$$N(L) = k \cdot L^d \quad (6)$$

En sådan funktion er skalainvariant, dvs. hvis vi omdefinierer "enheden" for længden L ved at gange talværdien med en vilkårlig positiv skalakonstant, ξ , så vil der kunne bestemmes en skalakonstant, n , for "massen" N , således at de skalerede størrelser

$$L_\xi = \xi \cdot L ; N_n = n \cdot N \quad (7)$$

stadig følger samme potenslov:

$$N_n(L_\xi) = k \cdot L_\xi^d \quad (8)$$

Vi skal blot vælge $n = \xi^d$.

Blandt "pæne" (analytiske, dvs. vilkårligt ofte differentiable) funktioner er det kun potensfunktioner, som således er invariante overfor skalatransformationer. At massefordelingen i en DLA struktur er skalainvariant betyder altså, at den ser ens ud i det store og i det små, og at den kan beskrives med en fraktal dimension d , som i ligning (6). Hvis længdeskalaen forøges med faktoren ξ , vil masseskalaen forøges med faktoren $n = \xi^d$, uanset hvor stor,

længdeskalaen var til at begynde med. Den fraktale dimension hænger altså sammen med skalafaktorerne ved formlen:

$$d = \log n / \log \varepsilon \quad (9)$$

Denne skalainvarians findes ofte for naturlige fænomener, men kun inden for et vist interval af størrelsesordener (med "en størrelsesorden" forstås normalt en faktor 10, dvs. 1 på en logaritmisk skala, hvis man bruger 10-tals logaritmer). På fig. 8 ser man kun en omtrentlig skalainvarians over knapt to størrelsesordener, og det er jo ikke meget, men skærmens begrænsede opløsningsevne levner jo ikke plads til ret meget mere.

En uendeligt udstrakt fraktal struktur, hvis dimension d er mindre end dimensionen af det euklidiske rum, den er indlejret i, vil have den gennemsnitlige massefylde 0 (og støvfnug er jo som bekendt lette), for massefylden er jo forholdet mellem masse og "rumfang" (areal i 2 dimensioner), og når massen N vokser med en mindre potens af den lineære dimension L end rumfanget, når L går mod uendelig, vil massefylden gå mod nul. Det er det samme forhold, som gør sig gældende for Serpinski-trekanten i foregående kapitel, som jo også viste sig at være uendelig let.

Skalainvariansen af DLA-støvfugget er en form for selv-similaritet, dvs. delen er ligedannet med det hele, men ligedannetheden er i dette tilfælde statistisk, ikke eksakt. På tilsvarende måde er selv-similariteten af naturlige former som bjerge, åer og skyer af statistisk art, men i enkelte tilfælde, som f.eks. bregnebladet, finder vi en selv-similaritet, som er næsten lige så eksakt som Serpinski-trekantens. I det følgende skal vi se på, hvordan vi konstruerer eksakt selv-similære former. Det sker ved indlejring af et formelement i sig selv, og metoden kaldes

rekursiv programmering Princippet kan sammenlignes med en kinesisk æske. Når vi åbner en sådan, finder vi en mindre æske indei, og når vi åbner denne, viser den sig at indeholde en endnu mindre æske. Således kan vi fortsætte, indtil vi kommer til den inderste æske, som ikke kan åbnes. Hvis vi har åbnet n æsker, før vi kommer til den inderste, siger vi, at den yderste æske er på niveau n , den næstyderste på niveau $n-1$, osv., den inderste på niveau 0.

Lad os forestille os, at vi er ude på tur med vores rumskib. Vi ser i det fjerne en underlig struktur:

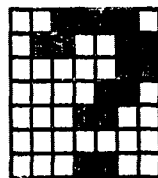
Den struktur, som vi ser i det fjerne, er en rekursiv struktur. Den består af mange små strukturer, som hver især består af endnu mindre strukturer. Dette er et eksempel på rekursiv programmering, hvor en struktur er defineret i termer af sig selv.

Fig. 9 . Hvad er dette?

På lang afstand (og det er mange lysår!) ligner det et spørgsmålstegn, opbygget af prikker. Mon disse prikker er stjerner? Nej, når vi kommer nærmere, ser vi, at de er spørgsmålstegn, opbygget af mindre prikker. Og selv disse prikker viser sig i kikkerten som små spørgsmålstegn, opbygget af prikker. Eller er det prikker?

Når vi er kommet så langt i vores overvejelser, har vi indset, at der er tale om en prikfraktal, dvs. en selv-similær struktur, som kan opbygges af en generator bestående af et prik-mønster. Generatoren for fig. 9 er åbenbart et spørgsmålstegn, som jo kan opbygges af prikker på et kvadratisk gitter. Bogstaver og andre tegn på computerens skærm kan opbygges på et kvadrat med 7×7 felter, og spørgsmålstegnet er opbygget som vist på fig. 10. Det må være tilstrækkelige data til konstruktion af det fraktale spørgsmålstegn i så mange niveauer, som der er plads til på skærmen. Når vi kender den lineære udstrækning af generatoren i forhold til byggeelementet (prikken), har vi skalafaktoren ξ i ligning (9), og antallet af prikker giver masse-skalafaktoren m . I dette tilfælde er $\xi=7$ og $m=16$, så vi kan straks udregne den fraktale dimension af strukturen på fig. 9 til $\log(16)/\log(7) = 1.42$.

Rekursiv prikfigur
Mode 1 (320*200) eller 2 (640*600)? 2
kantlængde (ulige, mellem 3 og 15)? 7



dimension 1.42
niveau 3

Fig. 10 . Generator for fig. 9.

Konstruktionen af en prikfraktal er simplest, hvis generatoren er bygget på et kvadratisk gitter med et ulige antal felter på hver led. Så er centret af kvadratet jo sammenfaldende med centret af det midterste felt. Vi kan angive koordinaten af midterfeltet i forhold til generator-kvadratet som (0,0). Lad os sige, at kantlængden af generatoren er det ulige heltal $g = 2s+1$, og at der indgår p prikker i generatoren. Koordinaterne for disse prikker kan så dimensioneres som arrays $xg(i)$ og $yg(i)$, hvor i løber fra 1 til p , og hvor hvert xg og yg kan antage værdier fra $-s$ til s . Alle disse indledende manøvrer kan ordnes simpelt med følgende lille program, som også udregner dimensionen:

```
DIM xg(49),yg(49)
INPUT "kantlængde (ulige) ": g
INPUT "antal prikker (max. 49) ": p
PRINT "dimension: ", LOG(p)/LOG(g)
FOR k:=1 TO p DO
    PRINT k;": ";
    INPUT "xg, yg ": xg(k),yg(k)
ENDFOR k
```

Selve konstruktionen varetages af en rekursiv procedure, dvs. en procedure, der kalder sig selv. Hvis proceduren udefra kaldes på niveau n , så må de i procedurekroppen forekommende kald være på et lavere niveau ($n-1$ eller mindre) og desuden må der være en konkret (ikke-rekursiv) beskrivelse af, hvad proceduren skal gøre, hvis niveauet er 0. Det kommer således til at foregå ligesom nedenstående, velkendte rekursive definition af fakultetsfunktionen:

$$n! = n \cdot (n-1)! ; 0! = 1. \quad (10)$$

Når en prikfraktal skal tegnes på niveau 0, skal der simpelthen sættes en prik i et opgivet punkt, (cx, cy) . For at udnytte skærmen bedst muligt, vil vi lade denne prik være så lille som muligt, dvs. en pixel. På niveau 1 vil vi lade centret ligge i det opgivne punkt (cx, cy) , og figurens udstrækning bliver g pixels på hver led. På niveau 2 skal vi for hver prik i generatoren tegne selve generatoren, og figurens udstrækning bliver g^2 pixels på hver led. Lad os nu se, hvordan vi kan opbygge proceduren:

```
PROC rekprik(n,cx,cy)
  IF n=0 THEN
    plot(cx,cy)
  ELSE
    FOR k:=1 TO p DO
      rekprik(n-1,cx+g↑(n-1)*xg(k),cy+g↑(n-1)*yg(k))
    ENDFOR k
  ENDIF
ENDPROC rekprik
```

Som vi skal se i næste kapitel, kan vi opbygge alle de rekursive tegneprocedurer efter samme enkle skema:

IF niveau=0 THEN tegn byggeelement ELSE opbyg generator med byggeelementer erstattet af den ønskede rekursive figur på et niveau, som er 1 mindre end det opgivne niveau.

For at få plads til den størst mulige rekursive prikfigur på skærmen må man lade centret (cx, cy) ligge midt på skærmen (320,100). Det maximale niveau, n_{max} , er så givet ved, at kantlængden g opløftet til potensen n_{max} skal være mindre end antallet af pixels på skærmens korte led (200). Dvs. vi kan benytte følgende procedurekald:

```
rekprik(INT(LOG(200)/LOG(g)),320,100)
```

Prøv nu at tegne nedenstående figur (Fournier-gitter på niveau 4). Her er $g=3$, $p=5$, dvs. dimensionen er $\log 5 / \log 3 = 1.46$. Koordinaterne for de 5 prikker i 3×3 generatoren er: $(-1,1)$, $(1,1)$, $(0,0)$, $(-1,-1)$ og $(1,-1)$.

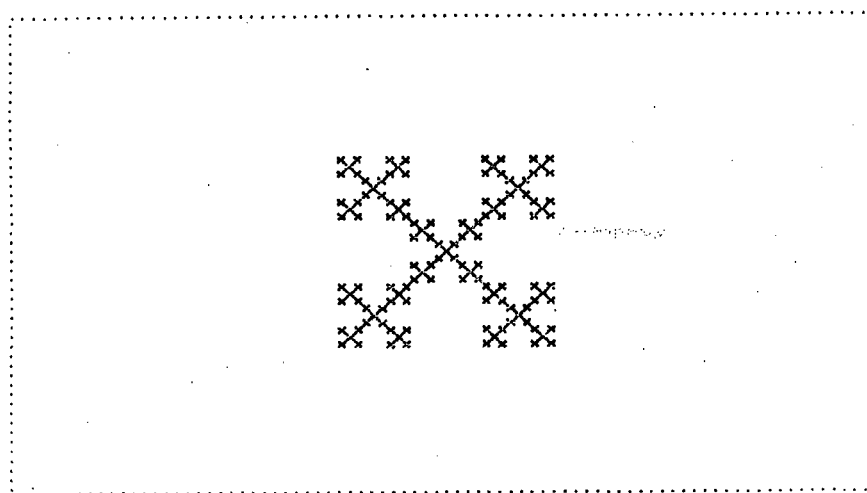


Fig. 11 . Fournier gitter, niveau 4.

Prøv også med $g=3$ at lave figuren med $p=2$, og koordinaterne $(-1,0)$ og $(1,0)$. Da punkterne her ligger på en vandret linje, og da skærmen har flere punkter på den vandrette led, er det muligt at gå til niveau 5. I dette tilfælde er dimensionen mindre end 1 ($\log 2 / \log 3 = 0.63$). Figuren kaldes det originale Cantor-sæt.

Kapitel 4 :

Rekursive stregtegninger: Skildpadde-generatorer

Prikfraktalerne i kapitel 3 var definerede ved en generator, hvis byggeelementer var prikker. På lignende måde kan vi definere stregfraktaler, hvor byggeelementet i generatoren er streger. Vi vil holde os til tilfælde, hvor generatoren er opbygget af lige lange linjestykker, og for at undgå trigonometriske udregninger af endepunkterne for disse linjestykker, vil vi benytte os af den såkaldte skildpadde-grafik, der findes som en særlig pakke i Uni-Comal og fremkaldes med kommandoen USE turtle. (RC-Comal har ikke turtle-pakken, men den findes på den til denne tekst hørende RC-Comal diskette).

Når skildpaddegrafikken er kaldt frem, ser man i et grafisk vindue midt på skærmen skildpadden som en pilespids, der peger lodret opad. Dette er skildpaddens udgangsposition, og til dette sted og denne retning vil den returnere, når den får kommandoen home. Øverst på skærmen er der et textvindue, hvor kommandoerne kan skrives, og man kan da straks se dem blive udført. Prøv f.eks. at skrive: ht (hide turtle, skildpadden bliver usynlig), st (show turtle, den kommer frem igen), rt(90) (right 90, drej 90° til højre), fd(100) (for~~ward~~ 100, gå fremad stykket 100), lt(45) (left 45, drej 45° til venstre), pu (pen up, løft pennen, så bevægelsen ikke trækker en streg), bk(50) (bak 50), pd (pen down, sænk pennen, klar til at tegne igen), home, cs (clear screen, slet grafiskskærmen).

Fordelen ved skildpadden fremfor den sædvanlige grafik-cursor er, at skildpadden husker sin retning, ikke blot sin position. Mange af kommandoerne virker således relativt til både den nuværende retning og position, f.

eks. fd(s), som flytter skildpadden fremad stykket s fra det punkt, hvor den stod, og i den retning, dens næse peger. Der findes dog også absolutte kommandoer, som f. eks. seth(v) (setheading), der indstiller retningen til vinklen v, regnet med uret fra lodret, uanset hvad retningen var før.

Den følgende tabel giver en oversigt over de vigtigste skildpadde-kommandoer, samt deres forkortelser. Ud over disse kan man stadig bruge de sædvanlige grafikkommandoer, såsom plot, draw og move.

Kommando.	Betydning.	Forkortelse.
back(s)	bak stykket s	bk(s)
clearscreen	slet grafik	cs
forward(s)	fremad stykket s	fd(s)
hideturtle	gem skildpadden	ht
home	gå hjem	
left(v)	drej v grader til venstre	lt(v)
pendown	sæk pen	pd
penup	løft pen	pu
right(v)	drej v grader til højre	rt(v)
setheading(v)	drej til retning v grader (fra lodret op, med uret)	seth(v)
showturtle	vis skildpadden	st

Kommandoen textscreen fjerner skildpaddegrafikken, så når man vil have den tilbage, må man igen skrive USE turtle.

Som en blød start vil vi nu skrive en skildpadde-procedure, som definerer en ny udgangsposition "ude vestpå med næsen mod øst".

```
PROC vest(s)
  seth(90)
  pu
  bk(s)
  pd
ENDPROC vest
```

Da skærmen har flest punkter på den vandrette led, er det bekvemt at lade de følgende stregfraktaler starte i et punkt A i venstre halvdel af skærmen og ende i et punkt B i højre halvdel. Vi vil forudsætte, at skildpadden før tegning står i A med næsen i retning af B, og at den slutter i B med næsen samme vej som ved starten. Som start kan man da skrive kommandoerne `home, cs, vest(100)`.

Vi skal nu skrive procedurer for nogle rekursive stregtegninger, hvis generatorer er vist på fig. 12.

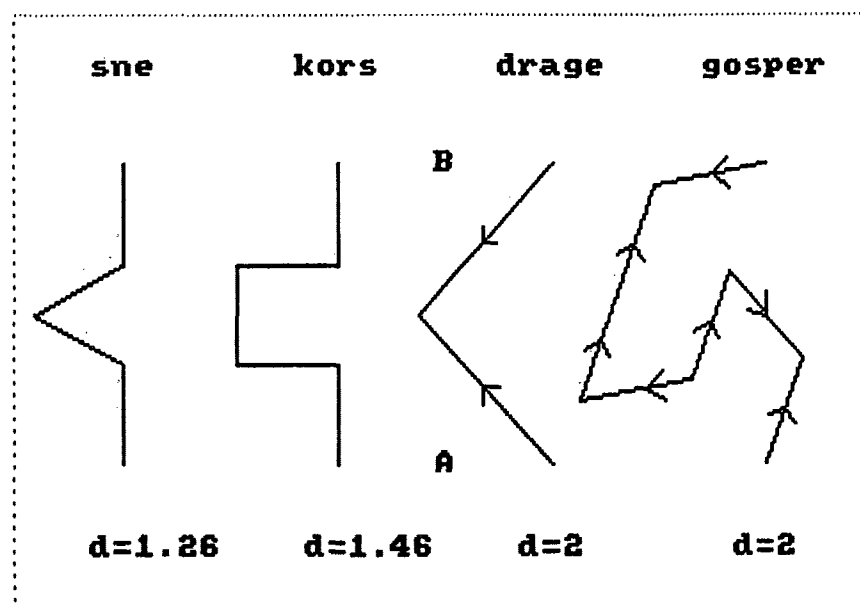


Fig. 12 . Generatorer for rekursive stregtegninger.

Generatorerne (niveau 1) for de rekursive tegninger er vist med begyndelsespunktet A nederst og slutpunktet B øverst. På niveau 0 vil de alle være en lodret streg fra A til B. Ud fra generatorerne kan vi udregne de fraktale dimensioner for figurerne efter samme system som tidligere. Lad os sætte længden af de enkelte linjestykker i generatoren til 1, og lad L betegne længden fra A til B, målt med denne enhed, medens N er antallet af linjestykker, der indgår i generatoren. L og N svarer så til de tidligere indførte skalafaktorer ξ og n , så dimensionen er givet ved:

$$d = \log N / \log L \quad (11)$$

Nedenstående tabel viser N og L og den fra (11) udregnede fraktale dimension for de af generatorerne på fig. 12 frembragte stregfraktaler:

	sne	kors	drage	gosper
N	4	5	2	7
L	3	3	$\sqrt{2}$	$\sqrt{7}$
d	1.26	1.46	2	2

To af fraktalerne, sne og kors, er altså "tynde", dvs. deres dimension er mindre end skærmens (2), medens de to andre, drage og gosper, er "fede" fraktaler" eller "space filling curves" som tenderer mod at udfylde et areal.

Vi starter med at lave en rekursiv skildpadde-procedure, der tegner kurven "sne" (den triadiske Koch-kurve eller snowflake-kurven). Vi skal bruge to input-parametre, niveaue n og AB-afstanden s . Hvert af de i i generatoren indgående linjestykker har så længden $s/3$, og disse linjestykker skal i den rekursive procedure erstattes med "sne" på niveau $n-1$, ifølge den sædvanlige opskrift:


```
PROC sne(n,s)
  IF n=0 THEN
    fd(s)
  ELSE
    sne(n-1,s/3)
    lt(60)
    sne(n-1,s/3)
    rt(120)
    sne(n-1,s/3)
    lt(60)
    sne(n-1,s/3)
  ENDIF
ENDPROC sne
```

Proceduren forudsætter, at skildpadden ved kaldet er i punktet A med næsen i retning af B, og at den slutter i B med næsen samme vej. Den sidste betingelse er konsistent med, at skildpadden enten går lige frem (hvis $n=0$) eller, at den drejer ialt 120° mod venstre og 120° mod højre. Det er ligeledes forudsat, at pennen er nede ved kaldet, og det vil den så vedblive at være under udførslen.

Hvis vi nu, når proceduren er kompileret (run) giver kommandoerne USE turtle, vest(100) og sne(1,200), skulle vi gerne se generatoren blive tegnet vandret hen over skærmen. Derefter kan vi prøve at tegne kurven på niveau 4. Også højere niveauer kan prøves, men man må betænke, at for hver gang, vi sætter niveauet 1 i vejret, stiger udførselstiden med en faktor 4, og længden af de enkelte linjestykker bliver hurtigt mindre end skærmens opløsningsevne.

Vi kan også prøve at lade proceduren sne indgå i et program som det følgende, der på niveau 1 tegner en sekstakket stjerne og på højere niveauer noget i retning af en snekrystal, som har givet snowflake kurven sit navn:

```
INPUT "kantlængde ": L
INPUT "niveau ": niv
USE turtle
graphicscreen(0)
pu
fd(L*0.29)
vest(L/2)
sne(niv,L)
rt(120)
sne(niv,L)
rt(120)
sne(niv,L)
```

Prøv nu på egen hånd at skrive en tilsvarende rekursiv procedure for kurven "kors". Denne kurve er beslægtet med det på fig. 11 viste "Fournier gitter", hvilket jo antydes af, at den har samme dimension ($\log 5 / \log 3$). Analogien viser sig klart, hvis man med kors-kurven vandrer rundt i et kvadrat mod uret. Hvis man går den anden vej rundt, kan man lave en flot "dækkeserviet i korssting".

Vi vender os nu til de to "fede" fraktaler, dragekurven og gosperkurven. På generatorerne for disse to er der angivet pile, der kan fortolkes som tegneretningen. Hvis vi ser på generatoren for dragekurven, viser den, at vi først skal gå (eller "drage") fra A til et punkt C på midtnormalen mellem A og B. Dernæst skal vi springe ned til punktet B, og herfra skal vi igen "drage" op til C. Endelig skal vi, som det hele tiden er forudsat, ende nede i B med næsen i retningen fra A til B. Skildpadden må altså foretage nogle hop undervejs, dvs. bevæge sig med løftet pen, men den skal slutte med pennen nede. Da forholdet mellem generatorlængden og linjestykket for drage-generatoren er givet ved det irrationale tal $\sqrt{2}$, kan det lønne sig i forvejen at

udregne dette tal og indlægge det i en konstant sq2.

```
sq2:=SQR(2)
PROC drage(n,s)
  IF n=0 THEN
    fd(s)
  ELSE
    lt(45)
    drage(n-1,s/sq2)
    lt(90)
    pu
    bk(s/sq2)
    pd
    drage(n-1,s/sq2)
    pu
    bk(s/sq2)
    rt(135)
    pd
  ENDIF
ENDPROC drage
```

Hvis dragekurven tegnes, så forbindelseslinjen AB er vandret eller lodret, vil den komme til at bestå af vandrette og lodrette linjestykker på de lige niveauer, men af skrå linjestykker på de ulige niveauer. De skrå linjer er ikke så pæne på skærmens raster-gitter. En pæn dragekurve på et ulige niveau opnås, hvis AB danner vinklen 45° med vandret. På 13 vises en tegneserie med de lige niveauer fra 0 til 14. At kurven har den fraktale dimension 2, viser sig ved, at den danner en "øgruppe". Forbindelsen mellem denne øgruppe og den tidligere omtalte "drageø" viser sig, hvis man først tegner dragekurven på et ulige niveau (f.eks. 13) så retningen AB går 45° opad (som på fig. 6) og derefter

tegner den samme kurve baglæns fra B til A. Vi vender tilbage til diskussion af dragekurven og drageøen i næste kapitel.

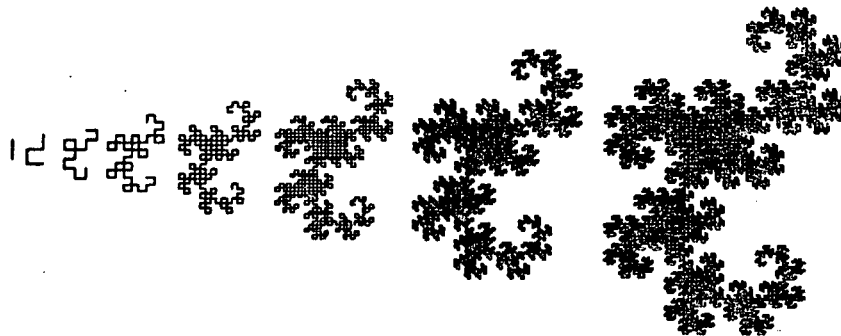


Fig. 13 . Dragekurven på de lige niveauer fra 0 til 14.

Vi går videre til Gosper-kurven, som er en af de mest interessante "space filling curves", fordi den, selv om den tenderer mod at udfylde et areal (kaldet "Frankrig"), på mirakuløs vis undgår at skære eller røre sig selv. Generatoren består af 7 linjestykker, af hvilke de to går i forlængelse af hinanden. Vinklerne mellem linjestykkerne er alle multipla af 60° , og afstanden mellem endepunkterne A og B er $\sqrt{7}$ gange så stor som længden af de enkelte linjestykker. Vi kan så ved brug af projektionssætningen udregne vinklen v mellem det første linjestykke og retningen AB. Vi skal blot løse ligningen

$$\begin{aligned} \cos(-v) + \cos(60^\circ - v) + \cos(180^\circ - v) + \cos(120^\circ - v) + 2 \cdot \cos(-v) + \\ \cos(-v - 60^\circ) = \sqrt{7} \end{aligned} \quad (12)$$

Ligning (12) har kun den ene løsning $\cos v = 5\sqrt{7}/14$, hvilket svarer til, at v må have værdien 19.1066 grader. Vi kan så lave gosper-proceduren, idet vi tager hensyn til de på generatoren angivne tegneretninger:

```
sq7:=SQR(7)
PROC gosper(n,s)
  IF n=0 THEN
    fd(s)
  ELSE
    rt(19.1066)
    gosper(n-1,s/sq7)
    lt(120)
    pu
    fd(s/sq7)
    rt(120)
    pd
    gosper(n-1,s/sq7)
    rt(120)
    gosper(n-1,s/sq7)
    rt(120)
    pu
    fd(s/sq7)
    pd
    gosper(n-1,s/sq7)
    rt(120)
    gosper(n-1,s/sq7)
    gosper(n-1,s/sq7)
    lt(120)
    pu
    bk(s/sq7)
    pd
    gosper(n-1,s/sq7)
```

```
    pu
    bk(s/sq7)
    rt(100.8934)
    pd
  ENDIF
ENDPROC gosper
```

Den sidste drejning til højre på 100.8934 grader er bestemt ved, at den samlede højredrejning skal være lig med den samlede venstredrejning, for at skildpadden kan slutte med næsen den rigtige vej.

Som det ses af fig. 14, ligger størstedelen af "Frankrig" over linjen AB, så for at få en stor figur, er det godt at starte under midten. Skriv f.eks. pu, bk(35), vest(75), gosper(4,150).

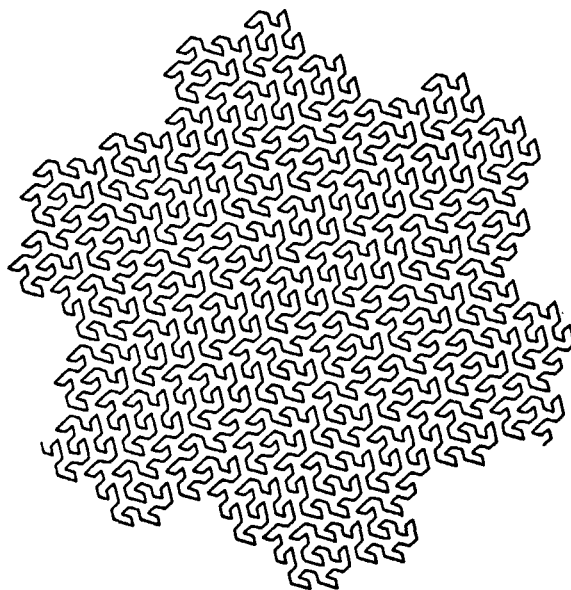


Fig. 14 . Gosperkurven, niveau 4.

Både drage- og gosperkurven er kontinuerte, så det kan virke lidt utilfredstillende at tegne dem så springende, som de to sidste procedurer gør. Kan de ikke tegnes, uden at pennen skal løftes undervejs? Jo, men for at gøre det, må vi have to procedurer, der gensidigt kalder hinanden. De dele af generatorerne, som gennemløbes "den forkerte vej" kan jo i stedet defineres som en anden procedure, hvor skildpaddens drejninger svarer til, at vi gennemløber samme kurve fra B til A. Nedenstående par af procedurer vil tegne gosperkurven sammenhængende:

```
PROC gosper1(n,s)
```

```
  IF n=0 THEN
```

```
    fd(s)
```

```
  ELSE
```

```
    rt(19.1066)
```

```
    gosper1(n-1,s/sq7)
```

```
    lt(60)
```

```
    gosper2(n-1,s/sq7)
```

```
    lt(120)
```

```
    gosper2(n-1,s/sq7)
```

```
    rt(60)
```

```
    gosper1(n-1,s/sq7)
```

```
    rt(120)
```

```
    gosper1(n-1,s/sq7)
```

```
    gosper1(n-1,s/sq7)
```

```
    rt(60)
```

```
    gosper2(n-1,s/sq7)
```

```
    lt(79.1066)
```

```
  ENDIF
```

```
ENDPROC gosper1
```

```
PROC gosper2(n,s)
```

```
  IF n=0 THEN
```

```
    fd(s)
```

```
  ELSE
```

```
    rt(79.1066)
```

```
    gosper1(n-1,s/sq7)
```

```
    lt(60)
```

```
    gosper2(n-1,s/sq7)
```

```
    gosper2(n-1,s/sq7)
```

```
    lt(120)
```

```
    gosper2(n-1,s/sq7)
```

```
    lt(60)
```

```
    gosper1(n-1,s/sq7)
```

```
    rt(120)
```

```
    gosper1(n-1,s/sq7)
```

```
    rt(60)
```

```
    gosper2(n-1,s/sq7)
```

```
    lt(19.1066)
```

```
  ENDIF
```

```
ENDPROC gosper2
```

På fuldkommen tilsvarende måde kan vi lave dragekurven

sammenhængende. Vi vil dog nu prøve at gøre det lidt anderledes, for forskellen på at gennemløbe dragekurven fra A til B eller fra B til A består jo i det væsentlige i, om man starter med at dreje 45° til venstre eller til højre. Denne forskel kan vi udtrykke ved at indføre en ekstra input parameter "op", som skal være +1, hvis vi starter med at dreje til venstre og -1, hvis vi drejer til højre. Vi kan så skrive $lt(op*45)$, for at dreje -45° til venstre er det samme som at dreje 45° til højre.

Medens vi er i gang, kan vi også indføre vinklen som en ekstra parameter v, så vi ikke altid skal dreje 45° . Afstanden mellem A og B er så $2 \cdot \cos v$, så dimensionen bliver

$$d = \log 2 / \log(2 \cdot \cos v) \quad (13)$$

Med den således generaliserede drageprocedure (dragegen, nedenfor) kan vi altså lave fraktaler med alle mulige dimensioner mellem 1 og 2 ved at lade v variere mellem 0° og 45° . For $v > 45^\circ$ bliver $d > 2$ iflg. (13) (og for $v = 60^\circ$, det skal man ikke tage for højtideligt). I proceduren benyttes en konstant $omr = \pi/180$, som omregner fra grader til radianer.

```
PROC dragegen(n,s,v,op)
  IF n=0 THEN
    fd(s)
  ELSE
    lt(op*v)
    dragegen(n-1,s/(2*COS(omr*v)),v,1)
    rt(2*op*v)
    dragegen(n-1,s/(2*COS(omr*v)),v,-1)
    lt(op*v)
  ENDIF
ENDPROC dragegen
```


I dragegen indgår to kald af dragegen, det første med $op=1$, det andet med $op=-1$. Som en sidste lille øvelse i dette afsnit kan man prøve at eksperimentere med ændringer af op -værdierne i disse kald. F.eks. kan man prøve at lade den være $+1$, eller -1 , eller op i begge kald. Eller man kan lave tilfældige, rekursive kurver, en slags brownske bevægelser med variabel dimension (den sædvanlige brownske bevægelse har fraktaldimensionen 2) ved at lade værdien af parameteren op , $+1$ eller -1 , i de to kald være tilfældigt bestemt. Fordelen ved at lave tilfældige kurver på denne måde er, at man ikke alene har styr på, hvor de starter, men også på, hvor de ender.

```
dimension 1.2  
niveau 10  
afstand 400  
længde 1270
```

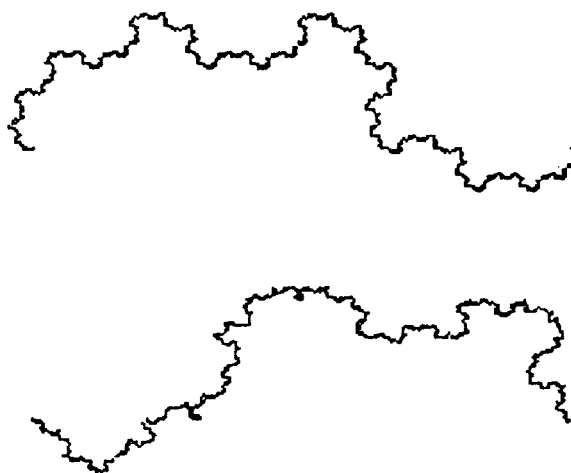


Fig. 15 . Øverst: regulær dragegen-kurve med $d = 1.2$.
Nederst: tilfældig kurve med samme dimension.

Kapitel 5 :

Dragekurven: Simple talforhold og puslespil .

I kapitel 2 fremstillede vi drageøen ved en meget simpel iteration, hvori der indgik et tilfældigt valg mellem to lineære funktioner. I kapitel 4 opdagede vi, at en todimensional fraktal kurve, dragekurven, fremstillet rekursivt med en uhyre simpel generator, er beslægtet med drageøen: Hvis vi først laver dragekurven fra A til B og derefter tilbage fra B til A, får vi drageøen. Denne overensstemmelse findes strengt taget kun "på niveau 0" for dragekurven, så når vi i det følgende taler om kurven uden at specificere niveauet, forudsætter vi stiltiende, at niveauet er højt nok til, at vi ikke i praksis har noget ud af at gå højere op. For sædvanlig skærmgrafik ligger denne grænse omkring niveau 15.

Den iterativt fremstillede drageø (fig. 6) har nogle bestemte koordinater: centret ligger i (0,0), punkterne A og B i hhv. (-1,-1) og (1,1). I dette koordinatsystem er afstanden AB altså lig med $\sqrt{8}$. Det omskrevne rektangel på fig. 6 rækker fra $x=-7/3$ til $x=7/3$ og fra $y=-5/3$ til $y=5/3$, og det indskrevne rektangel rækker fra $x=-5/3$ til $x=5/3$ og fra $y=-1/3$ til $y=1/3$. Vi kan anvende drageøen som puslespilsbrik, således at vi uden om den centrale ø med centrum i (0,0) anbringer fire andre brikker, hvis omskrevne rektangler berører det indskrevne rektangel for den centrale brik. Afstanden mellem centrerne for disse brikker må altså være $5/3 + 7/3 = 4$ i x-retningen og $1/3 + 5/3 = 2$ i y-retningen. Hele planen kan dækkes med sådanne brikker, som altså hver især råder over et rektangel med kantlængderne 4 og 2. Det følger heraf, at drageøen har arealet 8, dvs. (mere generelt) samme areal som et kvadrat med siden AB.

Som det fremgår af fig. 16 har hver brik i dette puslespil i virkeligheden 6 nærmeste naboer, som berører den. Tre farver er nødvendige, hvis man vil undgå, at brikker med samme farve støder sammen.

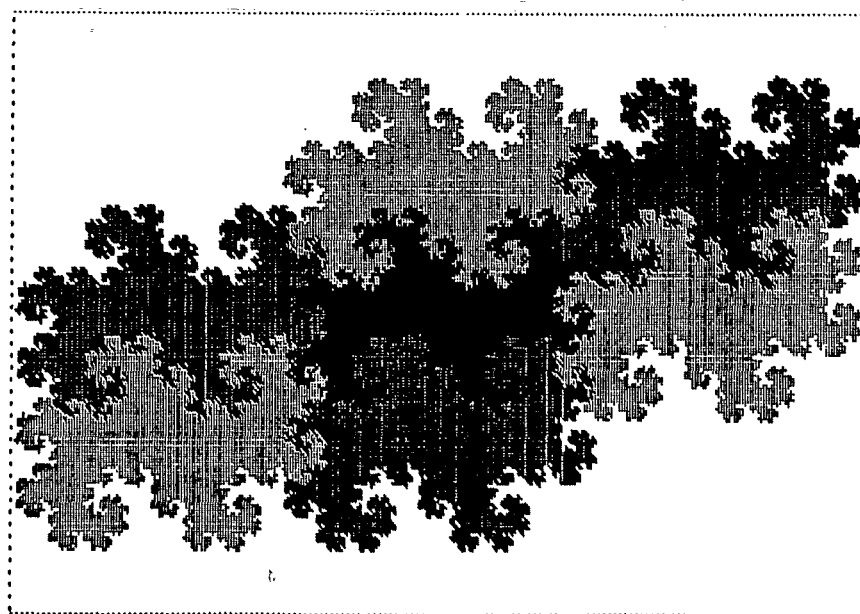


Fig. 16 . Drageøen som puslespilsbrik.

En dragekurve, som starter i $A=(0,0)$ og ender i $B=(1,1)$ har således arealet 1. Dette areal er summen af arealerne af de uendelig mange småøer, som dragekurvens øgruppe er sammensat af. Som det ses på fig. 17, er der én af disse øer, som er større end alle de andre; lad os kalde den "Sjælland" (denne ø, som indeholder punktet C, er tofarvet på fig. 17). Blandt de resterende øer er der igen én, som er større end de andre, nemlig nabøen til venstre for "Sjælland", lad os kalde den "Fyn". Bortset fra disse to, er de andre øer parvist lige store. Vi skal prøve at bestemme arealet af disse øer, først og fremmest "Sjælland" og "Fyn".

Først bemærker vi, at dragekurvens forløb fra A til B

kan beskrives med en reel parameter, t , som løber fra 0 til 1. En dragekurve på niveau n er jo sammensat af 2^n linjestykker, så vi kan fastsætte, at t skal være 0 ved tegningens start, og hver gang der er tegnet et linjestykke, skal der lægges 2^{-n} til t , så vil t have værdien 1, når tegningen er færdig. Dragekurven på niveau n er jo sammensat af to niveau $n-1$ kurver, den første fra A til C (mørkegrå på fig. 17), den anden fra B til C. Hvis kurven tegnes fortløbende (uden hop), hvilket vi vil forudsætte i det følgende, tegnes den anden niveau $n-1$ kurve fra C til B, og punktet C svarer til parameterværdien $t = 0.5$. Vi har så en kontinuert afbildning af intervallet $[0,1]$ på det af dragekurven beskrevne todimensionale område, og man kan let indse, at arealet af en ϕ er lig med længden af det til ϕ en svarende parameterinterval.

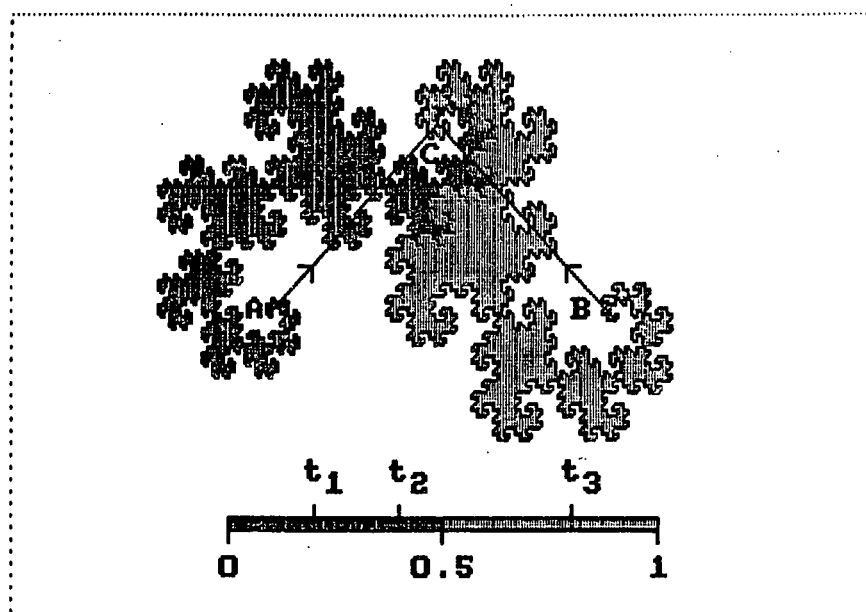


Fig. 17 . Dragekurven og parameterintervaller for ϕ er.

Proceduren, som har tegnet dragekurven på fig. 17 holder regnskab med parameteren t og kan derfor tegne de to halvdrager AC og CB med forskellig farve. Afstanden mellem A og B, målt i skærmkoordinater, er valgt til $2^{n/2}$, hvor n er niveauet, således at hvert linjestykke præcis svarer til én pixel. Dette muliggør, at dragekurvens omkreds tegnes med en anden farve end det indre. Det er nemlig sådan, at indre punkter besøges to gange, men ydre punkter kun én gang, så hvis computeren gør det muligt at teste, om et punkt tidligere er farvet, kan man give det en anden farve, når det besøges anden gang.

Lad antallet af indre punkter være n_i og antallet af ydre punkter n_y . Da det totale antal punktbesøg er 2^n , har vi:

$$2 \cdot n_i + n_y = 2^n \quad (14)$$

Det totale areal, målt med pixelarealet som enhed, vil være halvdelen af dette, hvilket stemmer med, at arealet skal være 1, når afstanden AB er $\sqrt{2}$: $\frac{1}{2} \cdot 2^n \cdot (\sqrt{2}/2^{n/2})^2 = 1$. De ydre punkter bidrager altså kun halvt så meget til arealet som de indre, men også kun halvt så meget til parameterens tilvækst. På tilstrækkeligt høje niveauer vil antallet af ydre punkter kun være en forsvindende brøkdel af antallet af indre punkter, og dette vil gælde for hvert endeligt parameterinterval, således at arealet af den tilsvarende del af kurven simpelthen bliver lig med parameterintervallets længde, når linjestykkerne AC og CB begge har længden 1.

Lad os sige, at øen "Fyn" svarer til parameterintervallet fra t_1 til t_2 på fig. 17 og "Sjælland" til intervallet fra t_2 til t_3 . Da "Fyn" i heldragen fra A til B er "Sjælland" i halvdragen fra A til C, har vi:

$$t_3 - t_2 = 2 \cdot (t_2 - t_1) \quad (15)$$

$$t_3 = 2 \cdot t_2 \quad (16)$$

Da den lysegrå halvdrage fra B til C er kongruent med den mørkegrå fra A til C, er det samlede areal af småøerne før "Fyn" identisk med arealet af småøerne efter "Sjælland", dvs.

$$t_1 = 1 - t_3 \quad (17)$$

Vi har så tre ligninger til bestemmelse af de tre parameter-værdier, og vi finder:

$$t_1 = 1/5 ; t_2 = 2/5 ; t_3 = 4/5 \quad (19)$$

"Fyn" har altså arealet $1/5$ og "Sjælland" $2/5$ af hele dragekurvens areal. Videre kan man så indse, at de to største af de parvist forekommende småøer hver er halvt så store som "Fyn", de to næststørste igen halvt så store, osv. og når vi regner det hele sammen:

$$\begin{aligned} &\text{Sjælland} + \text{Fyn} + 2 \cdot \text{småøer} = \\ &2/5 + 1/5 + 2 \cdot (1/10 + 1/20 + 1/40 + \dots) = 1 \end{aligned}$$

Når man således kender start- og slutværdien af parameteren t for hver af dragekurvens øer, kan man give dem forskellige farver og således skabe et nyt puslespil. Dragekurvens øer har ikke helt samme facon som den store, symmetriske drageø, men de muliggør en "tiling" af denne, som vist på fig. 18. Dette system kan så videreudbygges til flisedækning af hele planen. På fig. 19 vises et forslag til en ny type fliser på badeværelset, som dog nok ikke er så lette at sætte op som de almindelige, kvadratiske.

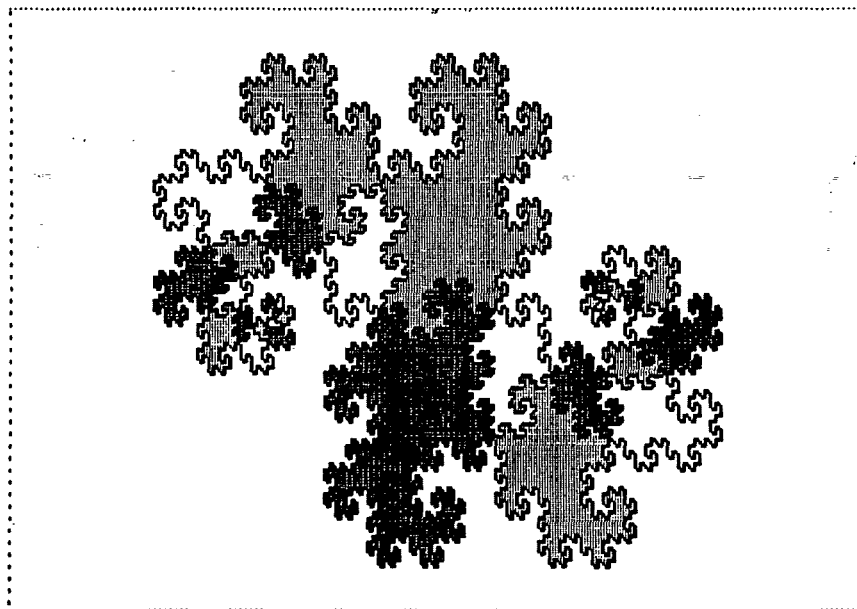


Fig. 18 . Tiling af drageøen med dragekurvens øer.

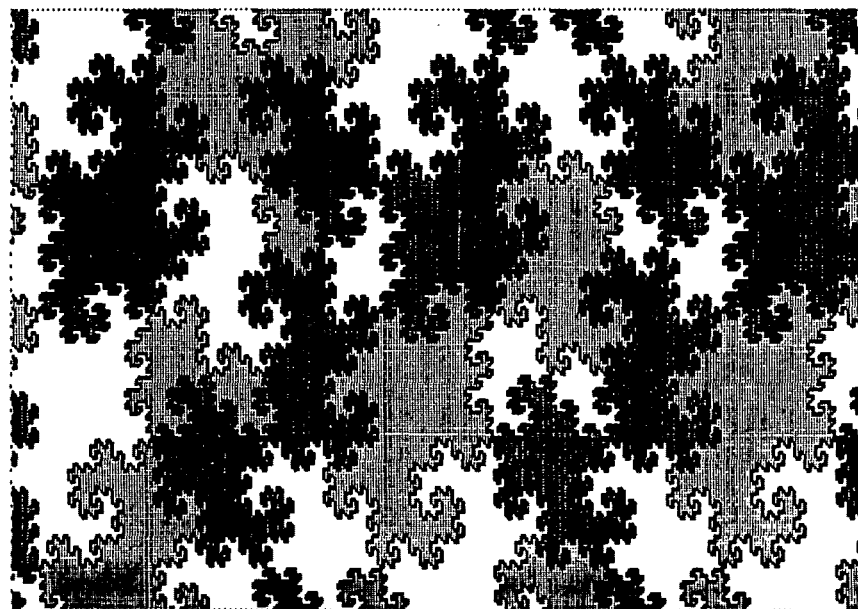


Fig. 19 . Et forslag til nye fliser på badeværelset.

Når vi nu har løst arealproblemet for drageøen, melder spørgsmålet sig: hvad er dens omkreds. Dette spørgsmål skal vi dog ikke forvente et fast tal som svar på, for omkredsen er en fraktal kurve, og længden af en sådan vokser eksponentielt med niveauet for en fast afstand mellem endepunkterne A og B. Hvis vi som før lader AB-afstanden $s(n)$ være $2^{n/2}$ (n er niveauet), således at de enkelte små linjestykker i dragekurven har længden én pixelafstand, kan omkredsen af hele drageøen udtrykkes på formen

$$n_y(n) = k \cdot (s(n))^d \quad (20)$$

hvor d er omkredsens fraktale dimension (sml. lign. (6)). Når vi så for fast n går over til standard-længdeenheden, således at længden AB er $\sqrt{2}$, finder vi omkredsens længde, $O(n)$, ved at dividere n_y med $s(n)$ og gange med $\sqrt{2}$, dvs.

$$O(n) = k' \cdot (\sqrt{2})^n \cdot (d-1) \quad (21)$$

hvor $k' = k \cdot \sqrt{2}$. Da d jo er større end 1 for en kontinuert fraktal kurve, ser vi altså, at omkredsen vokser eksponentielt med niveauet. Direkte optælling af n_y som funktion af n giver resultaterne

n	12	13	14	15	16	17
n_y	1202	2038	3457	5856	9938	16854

Vi kan da ved sammenligning med formlen (sml. (20))

$$n_y = k' \cdot (\sqrt{2})^n \cdot d-1 \quad (22)$$

bestemme det bedste fit for konstanterne k' og d. Resultatet er $k' = 3.0000$ og $d = 1.5242$.

En nøjere undersøgelse af vores andet eksempel på en "fed" fraktalkurve, gosperkurven, kan gennemføres på lignende måde. Her skal blot nævnes nogle resultater, som man eventuelt kan efterprøve. Når kurven tegnes fortløbende, kan man ved hjælp af en parameterfremstilling få en kontinuert afbildning af enhedsintervallet på arealet af "Frankrig". Hvert af de 7 linjestykker i generatoren svarer til en gosperkurve på niveau $n-1$, og hver af disse kan eventuelt gives sin egen farve ved tegningen. Som det fremgår af fig. 20, er "Frankrig" en puslespilsbrik til tiling af planen, som kan opdeles i 7 mindre brikker af samme form.

Hvis hvert af linjestykkerne i generatoren har længden 1 ($AB = \sqrt{7}$), er gosperkurvens samlede areal $7\sqrt{3}/2$. Ved at forlænge nogle af disse linjestykker på fig. 20 opstår der nogle ligesidede trekanter, og man kan ret nemt pusle sig frem til, at der skal 14 trekanter til for at dække arealet. Den fraktale dimension af omkredsen er ca 1.2.

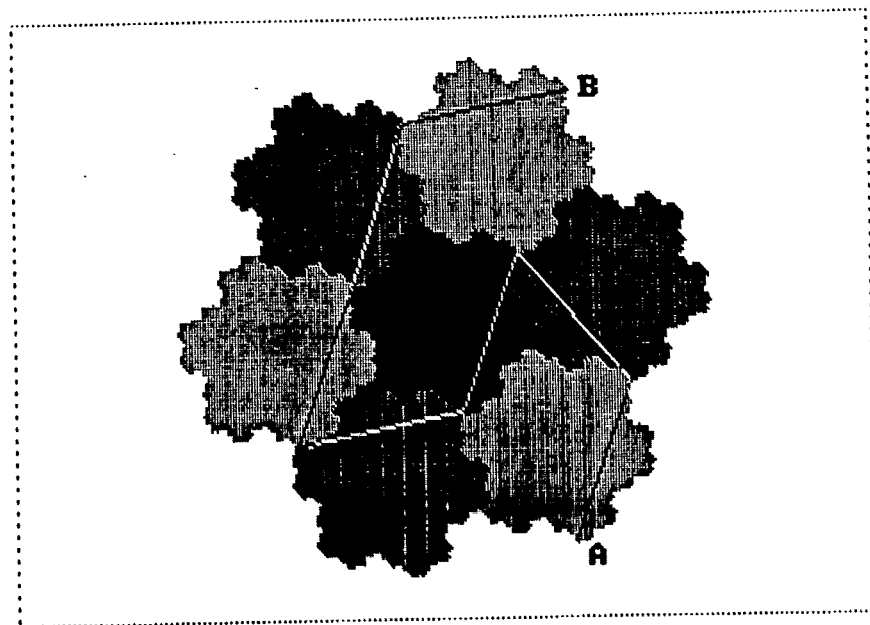


Fig. 20 . Gosperkurvens opdeling.

Kapitel 6 :

Refleksion fra cylindre .

I de sidste to kapitler skal vi se på et par simple fysiske fænomener med en fraktal og kaotisk opførsel. I modsætning til DLA modellen i kapitel 3, som var rent tilfældig, er der her tale om stærkt lovbunden adfærd, faktisk så lovbunden, at fænomenerne i deres renhed kun vanskeligt kan observeres i laboratoriet, men til gengæld så meget lettere på computerskærmens simulerede virkelighed.

Den første model er refleksion af en lysstråle fra en mængde spejlende cylindre . Opstillingen fremgår af fig. 20. I den højre del af figuren ses et todimensionalt snit af de cirkulære cylindre. Antallet af cylindre skal være mindst 3, hvis der skal komme noget spændende ud af eksperimentet.

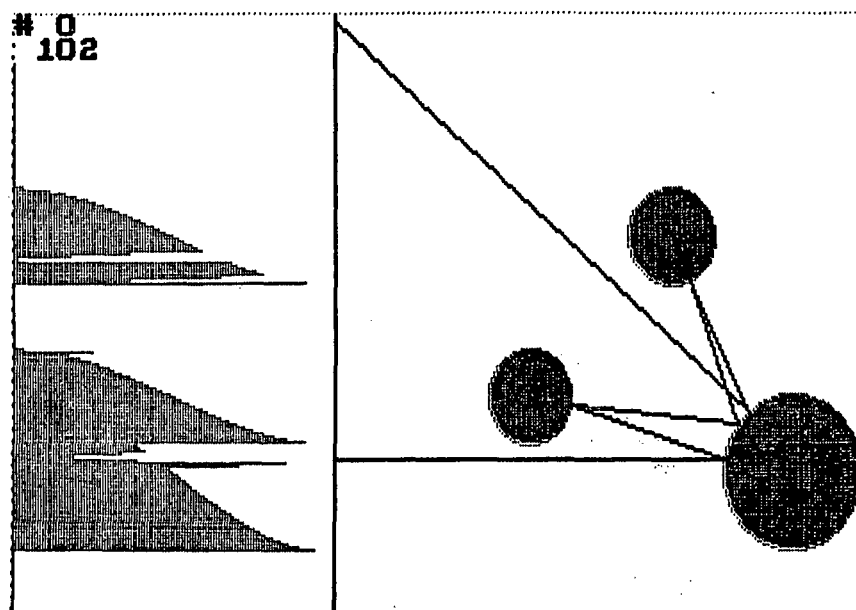


Fig. 21 . Refleksion af lysstråle fra cylindre.

Cylindrenes radier og placering på skærmen kan være tilfældig, blot må de ikke overlappe hinanden. En lysstråle sendes vandret ind fra venstre i en vis højde y . Efter et antal reflektioner (indfaldsvinkel = udfaldsvinkel), som kan variere fra 0 til ∞ , forlader strålen området igen, og den danner nu en vis vinkel v (mellem 0 og 2π) med vandret. Opgaven går ud på at afbilde v som funktion af y . En sådan afbildning ses i venstre del af fig. 20. Bortset fra "falske diskontinuiteter", hvor v springer fra 0 til 2π , vil der være skarpe spidser og kløfter ind imellem områder med glat (differentiabel) variation. Hvis man udvælger et y -område omkring en spids eller kløft, kan man i næste omgang lade y variere inden for dette område, men strække afbildningen af v , så den fylder hele den lodrette akse. Man vil så opdage ny struktur i dette område. I princippet kan man på denne måde forstørre kurven vilkårligt op. Det viser sig da, at den er fraktal i den forstand, at den ikke bliver glat, selv ved meget stor forstørrelse, men vedblivende afslører nye spidser og kløfter.

Lad os nu se, hvordan vi behandler systemet matematisk med henblik på skrivning af et program. Det drejer sig i første omgang om at bestemme skæringspunkter mellem en cirkel og en ret linje. I anden omgang skal vi sørge for at udvælge det rigtige skæringspunkt som det sted, hvor der sker reflektion, og endelig skal vi bestemme en ny ret linje, svarende til den reflekterede lysstråle. Når vi er nået så langt, gentager problemet sig, og vi skal blot blive ved med at løse det, så længe, der er en løsning, dvs. indtil lysstrålen slipper væk.

Hvis en lysstråle udgår fra punktet (x_0, y_0) og har retningsvinklen v_0 , kan den beskrives som en ret linje med parameterfremstillingen

$$x(t) = x_0 + t \cdot \cos(v_0) ; y(t) = y_0 + t \cdot \sin(v_0) \quad (23)$$

Antag, at vi har en cylinder med radius r liggende med centrum i (c_x, c_y) . Strålens vinkelrette afstand fra centret er da:

$$a = \text{ABS}(\cos(v_0) \cdot (c_y - y_0) - \sin(v_0) \cdot (c_x - x_0)) \quad (24)$$

En nødvendig betingelse for, at strålen rammer cylinderen er naturligvis, at a er mindre end r . Vi må imidlertid også forlange, at t -værdien for skæringspunktet skal være positiv (ellers er det jo strålens forlængelse bagud, der rammer), og når der så er skæring, skal vi vælge skæringspunktet med den mindste t -værdi. Når vi har udregnet a (CM på fig. 22) og b (QM), kan vi finde indfaldsvinklen (PQL) og derfra bestemme retningen for den reflekterede stråle (QR).

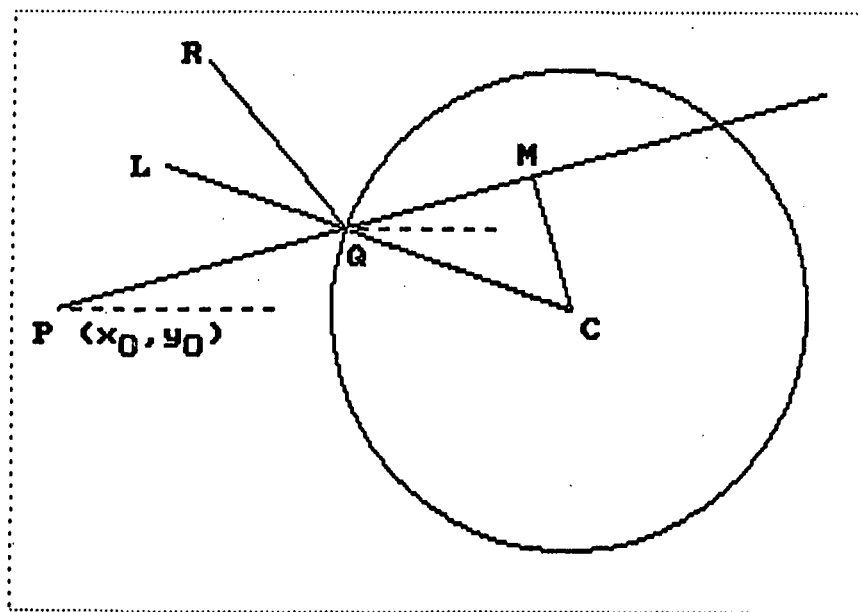


Fig. 22 . Strålegang PQR ved refleksion fra cylinder.

Nedenstående procedure ram(cx,cy,r) bestemmer skæringspunktet Q mellem en indgående stråle og en cirkel med centrum (cx,cy) og radius r og retningsvinklen i forhold til vandret for den reflekterede stråle. Linjens variable x0, y0 og t er eksterne (globale), dvs. defineret uden for proceduren. Hvis der ikke er noget skæringspunkt, tildeler proceduren t en negativ værdi:

```
PROC ram(cx,cy,r)
  cv:=COS(v0)
  sv:=SIN(v0)
  a:=ABS(cv*(cy-y0)-sv*(cx-x0))
  IF a>r THEN
    t:=-1000
  ELSE
    b:=SQR(r*r-a*a)
    t:=cv*(cx-x0)+sv*(cy-y0)-b
    IF t>0 THEN
      x0:=x0+t*cv
      y0:=y0+t*sv
      v0:=v0+PI-2*ATN(a/b)
      WHILE v0<0 DO v0:=v0+2*PI
      WHILE v0>2*PI DO v0:=v0-2*PI
    ENDIF
  ENDIF
ENDPROC ram
```

Proceduren bestemmer altså udgangspunktet (x0,y0) samt retningen for den reflekterede stråle, og herefter kan den så anvendes igen og igen, indtil den sidst reflekterede stråle ikke rammer nogen af cylindrene (t negativ). Problemet er ikke rekursivt, for der er ingen "bund" i det, svarende til niveauet 0. Hver gang en ny stråle er beregnet,

er situationen den samme som før, og i princippet kan der være uendelig mange refleksioner, men i praksis slipper strålen altid ud før eller senere.

Lad os antage, at cylindrenes centre og radier er indlagt i nogle arrays $cx(k)$, $cy(k)$ og $r(k)$, hvor k kan gå fra 1 til n . For en given højde y af den indkommende, vandrette stråle kan vi da beregne hele strålegangen ved følgende programstump:

```
x0:=200
y0:=y
v0:=0
t:=1
moveto(x0,y0)
WHILE t>0 DO
  k:=0
  t:=0
  WHILE t<=0 AND k<n DO
    k:=k+1
    ram(cx(k),cy(k),r(k))
  ENDWHILE
  IF t>0 THEN
    drawto(x0,y0)
  ELSE
    draw(50*cv,50*sv)
  ENDIF
ENDWHILE
```

Når (hvis) programmet slipper ud af den yderste WHILE-løkke, vil $v0$ være retningsvinklen for den udgående stråle, og denne vinkel kan så afbildes som funktion af y . Vi skal ikke her gå i flere detaljer med programmeringen, men henviser til disketten eller programlisten i appendix A.

Kapitel 7 :

Bold i grøft .

En fuldstændig elastisk bold slippes fra en vis højde h og falder ned i en grøft, hvis bund har højden 0. Her er den så dømt til for tid og evighed at springe frem og tilbage mellem grøftens to sider. Falde til ro i bunden kan den aldrig, for dens elasticitet sikrer jo, at summen af kinetisk og potentiel energi vedblivende vil være det samme som den potentielle energi i det øjeblik, vi gav slip på bolden, altså

$$E = \frac{1}{2}m \cdot (v_x^2 + v_y^2) + mgy = mgh \quad (25)$$

hvor (v_x, v_y) er boldens hastighedsvektor, y dens højde, m dens masse og g tyngdeaccelerationen.

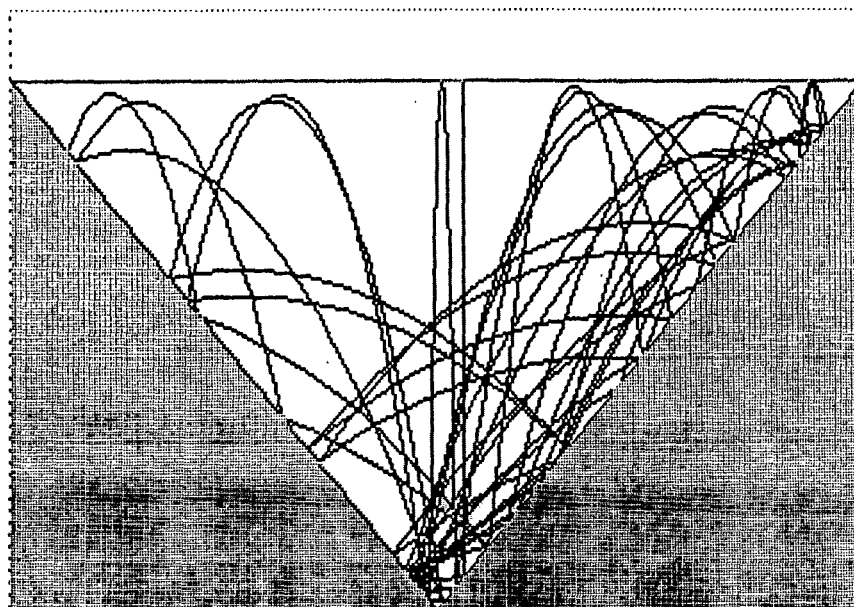


Fig. 23 . Banekurve for bold i grøft.

Vi beskriver grøften som to skråplaner med den numeriske hældningskoefficient α . Hvis grøftens bund vælges som nulpunkt for vores koordinatsystem, kan de to skråplaner beskrives ved ligningen

$$y = \alpha \cdot \text{ABS}(x) \quad (26)$$

For at undgå for mange konstanter, vil vi sætte boldens masse til 1. Ifølge Galileis faldlov er bevægelsen i tyngdefeltet jo uafhængig af massen, og vi kan så bruge bogstavet m til noget andet. Altså: lad os se på situationen lige efter, at bolden har foretaget det mte opspring fra et af de to skråplaner. Hvis x_m -koordinaten ved dette opspring er x_m , må hastigheden iflg (25) og (26) tilfredsstille betingelsen

$$v_m^2 = v_{xm}^2 + v_{ym}^2 = 2g \cdot (h - \alpha \cdot s_m \cdot x_m) \quad (27)$$

hvor $s_m = \text{SGN}(x_m)$ er fortegnet af x_m . Vi skal så beregne stedet og hastigheden for det $(m+1)$ te opspring. Vi kan sige, at opspring m sker til tiden $t=0$. Indtil bolden rammer igen, har vi så:

$$v_x(t) = v_{xm} ; v_y(t) = v_{ym} - g \cdot t \quad (28)$$

$$x(t) = x_m + v_{xm} \cdot t ; y(t) = y_m + v_{ym} \cdot t - \frac{1}{2}gt^2$$

Det næste nedslag af bolden finder sted, når $x(t) = x_{m+1}$, $y(t) = y_{m+1} = \alpha \cdot \text{ABS}(x_{m+1}) = \alpha \cdot s_{m+1} \cdot x_{m+1}$. Der er nu to tilfælde, vi må skelne imellem:

- 1) Bolden rammer samme skråplan: $s_{m+1} = s_m$.
- 2) Bolden rammer modsatte skråplan: $s_{m+1} = -s_m$.

Lad os først antage, at tilfælde 1) foreligger. Vi får da af (28):

$$y_m + v_{ym} \cdot t - \frac{1}{2} g t^2 = \cos \alpha \cdot s_m \cdot (x_m + v_{xm} \cdot t) \quad (29)$$

og da $y_m = \cos \alpha \cdot s_m \cdot x_m$, finder vi tiden for nedslag nr. $m+1$ til

$$t = \frac{2}{g} \cdot (v_{ym} - \cos \alpha \cdot s_m \cdot v_{xm})$$

og indsættelse af dette i (28) giver så

$$x_{m+1} = x_m + v_{xm} \cdot \frac{2}{g} \cdot (v_{ym} - \cos \alpha \cdot s_m \cdot v_{xm}) \quad (30)$$

Vi har her forudsat tilfælde 1), dvs. at x_m og x_{m+1} har samme fortegn. Så må det gælde, at

$$x_m \cdot [x_m + v_{xm} \cdot \frac{2}{g} \cdot (v_{ym} - \cos \alpha \cdot s_m \cdot v_{xm})] > 0 \quad (31)$$

Omvendt må vi kunne slutte, at hvis (31) ikke er opfyldt, så er det tegn på, at vi er i tilfælde 2), selv om udtrykket (30) i dette tilfælde ikke er rigtigt. Når (31) er opfyldt, bliver hastigheden ved nedslag nr. $m+1$ givet ved

$$v'_{x,m+1} = v_{xm} \quad (32)$$

$$v'_{y,m+1} = v_{ym} - 2 \cdot (v_{ym} - \cos \alpha \cdot s_m \cdot v_{xm}) = 2 \cdot \cos \alpha \cdot s_m \cdot v_{xm} - v_{ym}$$

Vi lader altså v' betegne hastighedskomponenter ved nedslag, medens v står for opspringet lige efter refleksionen. Selve refleksionen beskrives lettest, når vi indfører hastighedens komponenter v_t og v_n efter hhv. tangenten og normalen til skråplanen på det sted, hvor refleksionen skete.

$$v_x = \frac{1}{\sqrt{1+\alpha^2}} \cdot (v_t - \alpha \cdot s \cdot v_n) \quad (33)$$

$$v_y = \frac{1}{\sqrt{1+\alpha^2}} \cdot (\alpha \cdot s \cdot v_t + v_n)$$

Ved reflektionen sker der så simpelthen det, at v_n skifter fortegn, medens v_t bevarer sin værdi, dvs. når vi benytter (33) på hastigheden ved opspring m , indsætter dette i (32) og så skifter fortegn på v_n , får vi hastighedskomponenterne ved opspring nr. $m+1$, dvs. $v_{x,m+1}$ og $v_{y,m+1}$. Disse udtryk kan vi så indsætte i de omvendte formler til (33) og finde komponenterne efter tangenten og normalen ved opspring nr. $m+1$, stadig under forudsætning af tilfælde 1):

$$v_{t,m+1} = v_{tm} - 2\alpha s_m v_{nm} \quad (34)$$

$$v_{n,m+1} = v_{nm}$$

Simpelheden af disse udtryk viser, at det kan betale sig at bruge hastighedens komponenter efter tangent og normal ved opspringet som tilstandsvariable. Dette har den yderligere fordel, at vi så ved, at den ene variabel, v_n altid må være positiv. Energibevarelsen (25) giver

$$v_t^2 + v_n^2 = 2gh \quad (35)$$

dvs. punktet (v_t, v_n) må ligge inden for en halvcirkel med radius $\sqrt{2gh}$. Betingelsen (31) for, at vi er i tilfælde 1), kan omformuleres til

$$x_m \cdot \left[x_m + \frac{2}{g} \cdot v_{nm} \cdot (v_{tm} - \alpha \cdot s_m \cdot v_{nm}) \right] > 0 \quad (36)$$

Vi må nu behandle det vanskeligere tilfælde 2), når (36) ikke er opfyldt. Der skal så det modsatte fortegn på

højresiden af (29), og tiden fra det mte opspring til det (m+1)te nedslag findes til

$$t = \frac{1}{g} \cdot (\alpha s_m v_{xm} + v_{ym} + W_m)$$

hvor

$$W_m = \sqrt{(\alpha s_m v_{xm} + v_{ym})^2 + 4gy_m} \quad (37)$$

En lidt omstændelig udregning fører så til, at vi i tilfælde 2) i stedet for (34) får:

$$v_{t,m+1} = v_{t,m} - \alpha s_m v_{nm} + \frac{\alpha}{\sqrt{1+\alpha^2}} \cdot W_m$$

$$v_{n,m+1} = \frac{1}{\sqrt{1+\alpha^2}} \cdot W_m \quad (38)$$

Størrelsen W kan også udtrykkes ved v_t og v_n :

$$W^2 = 4gh - 2v_n^2 + \frac{1-\alpha^2}{1+\alpha^2} \cdot [(1-\alpha^2)v_n^2 - 2v_t^2 + 4\alpha v_t v_n] \quad (39)$$

Der er ét tilfælde, som er særlig simpelt, nemlig tilfældet $\alpha = 1$, hvor grøftens to sider står vinkelret på hinanden. Så længe bolden springer på samme side af grøften, har normalhastigheden jo en konstant værdi v_1 . Når den så springer over til den modsatte side, får den en ny værdi v_2 . I tilfældet $\alpha = 1$, hvor sidste led i (39) forsvinder, er værdien af v_2 uafhængig af tangentialhastigheden, og vi finder simpelthen, at normalhastighederne er de samme, hver gang bolden er på en bestemt side af grøften. De to værdier hænger sammen på følgende måde:

$$v_1^2 + v_2^2 = 2gh \quad (40)$$

Med brug af (34) eller (38), alt efter om (36) er opfyldt eller ej er problemet reduceret til en iteration af

samme type som beskrevet i kapitel 1. Hver gang, bolden har lavet et opspring, kan vi afsætte et punkt på skærmen, svarende til værdierne af v_t og v_n , og vi får da dannet en såkaldt Poincaré-afbildning. På fig. 24 ses en sådan afbildning for tilfældet $\alpha = 1.1$ og for tre forskellige værdier af $\xi = \alpha x_0/h$. For $\xi = 0.1$ ses en kaotisk bevægelse og for de to andre værdier to forskellige typer af næstenperiodiske bevægelser (perioder 3 og 5). Analogien med Martins formel er tydelig.

I tilfældet $\alpha = 1$ er alle bevægelser næstenperiodiske, og Poincaré-afbildningen fremviser to vandrette linjer, svarende til værdierne v_1 og v_2 . For $\alpha < 1$ er alle bevægelser kaotiske, og for $\alpha > 1$ findes en kompliceret blanding af kaotiske og næstenperiodiske bevægelser.

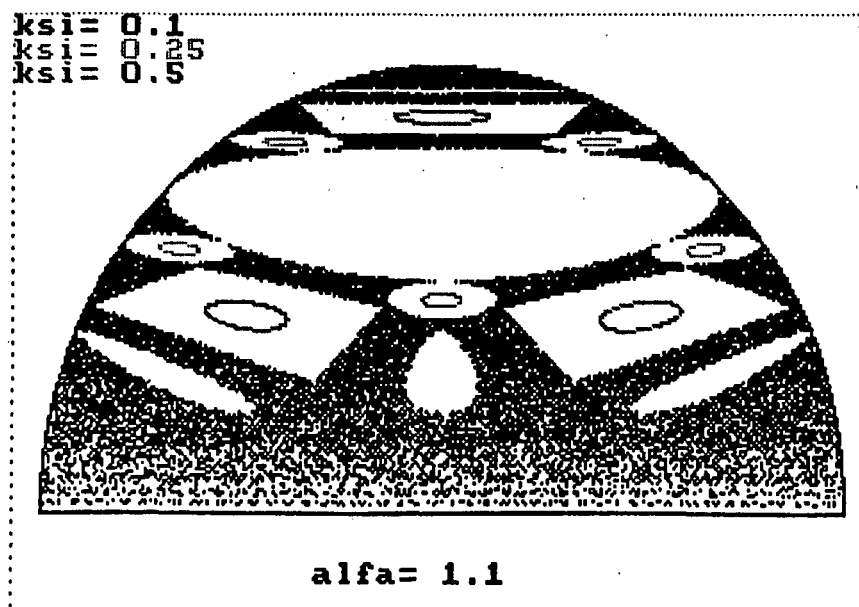


Fig. 24 . Poincaré-afbildning for den hoppende bold.

Appendix A :

Programliste, refleksion fra cylindre .

```
0010 // Program PERIFLEX.           Heine Larsen.   December 1988
0020 USE system
0030 USE graphics
0040 USE sound
0050 fullscreen
0060 aspect:=0.8 // inq(30)/inq(31)
0070 xpixels:=inq(20); ypixels:=inq(22)
0080 f$:=""18" -#.##### "19"
0090 DIM infline$ OF 50
0100 infline$:=""18"+18* " "+"19"
0110 ncylmax:=12
0120 ncyl:=3
0130 mr:=0.6
0140 sr:=0.2
0150 RANDOMIZE
0160 DIM r(1:ncylmax), cx(1:ncylmax), cy(1:ncylmax), a(1:ncylmax), b(1:ncylmax)
0170 selectwindow(2)
0180 DIM shape$ OF shapsize#(inq(19),inq(20),inq(21),inq(22))
0190 putstyle(0)
0200 info
0210 main
0220 textwindow(1,25,1,80)
0230 textscreen
0240 END
0250
0260
0270
0280 PROC info
0290  PAGE
0300  PRINT ""0,";
0310  PRINT AT 5,20: ""18" REFLEKSION FRA CYLINDRE "19"
0320  PRINT AT 8,5: "Programmet viser strålegangen for en lysstråle, som sendes ind fra
venstre";
0330  PRINT AT 9,5: "og reflekteres fra tilfældigt anbragte cylindre."
0340  PRINT AT 10,5: "I et diagram til venstre afsættes afbøjningsvinkel vandret og højde
lodret."
0350  PRINT AT 12,5: "Et nyt sæt cylindre vælges ved at trykke ""c"", og angive parametre.";
0360  PRINT AT 14,5: "Nederste linje angiver Y-min, Y og Y-max og indikationer for
tilstanden.";
0370  PRINT AT 14,5: "Et scan af Y-værdier startes og stoppes ved tryk på ENTER.";
0380  PRINT AT 15,5: "Lyden slås til og fra ved tryk på ""l"".";
0390  PRINT AT 16,5: "Ved at trykke ""z"" kan man zoome ind på diagrammet med"
0400  PRINT AT 17,5: "piletasterne, page up, page down, home og end. "
0410  PRINT AT 18,5: "Ved at trykke på TAB-tasten skifter man mellem øvre og nedre grænse"
0420  PRINT AT 19,5: "Zoom afsluttes ved tryk på ESC."
0430  PRINT AT 25,20: "Tryk på en tast ...";
0440  WHILE KEY$="" DO NULL
0450  PRINT ""0,";
0460  PAGE
0470 ENDPROC info
0480
0490 PROC main
0500  scanning:=FALSE
0510  pause:=FALSE
```

```
0520 lyd:=TRUE
0530 fast:=FALSE
0540 zooming:=FALSE
0550 ymin:=-2; ymax:=2
0560 cylindre
0570 ymin:=inq(25); ymax:=inq(26); nstep:=inq(22)-inq(21)
0580 ystart:=ymin; yslut:=ymax; ystep:=(yslut-ystart)/nstep
0590 reset_scan
0600 infoline
0610 LOOP

0620 IF scanning THEN
0630   y:=ynext; ynext:=y+ystep
0640   ray
0650   diagram
0660   IF y>yslut THEN
0670     scanning:=FALSE
0680     infoline
0690     y:=yslut; ynext:=ystart
0700   ENDIF
0710 ENDIF
0720 c$:=KEY$
0730 CASE c$ OF
0740 WHEN ""
0750 WHEN ""13""
0760   scanning:=NOT scanning
0770   infoline
0780 WHEN " "
0790   IF scanning THEN
0800     infoline
0810   ELSE
0820     selectwindow(2)
0830     IF fast THEN putshape(inq(19).inq(21).shape$)
0840     ray
0850   ENDIF
0860   infoline
0870 WHEN ""27""
0880   EXIT
0890 WHEN "1"
0900   lyd:=NOT lyd
0910   infoline
0920 WHEN "c"
0930   scanning:=FALSE
0940   cylindre
0950   ystart:=ymin; yslut:=ymax
0960   reset_scan
0970   infoline
0980 WHEN "+"
0990   fast:=TRUE
1000   infoline
1010 WHEN "-"
1020   fast:=FALSE
1030   infoline
1040 WHEN "1"
1050   getnumber("Y-Start = ", ystart)
1060   ystart:=minmax(ymin, ystart, ymax)
1070   IF ystart>yslut THEN swap(ystart, yslut)
1080   reset_scan
1090   infoline
1100 WHEN "2"
1110   getnumber("Y-Slut = ", yslut)
1120   yslut:=minmax(ymin, yslut, ymax)
1130   IF ystart>yslut THEN swap(ystart, yslut)
1140   reset_scan
1150   infoline
```

```
1160     WHEN "z"
1170         IF NOT scanning THEN zoom
1180         reset_scan
1190         infoline
1200     OTHERWISE
1210     ENDCASE
1220 ENDLOOP
1230 ENDPROC main
1240
1250 PROC ray
1260     selectwindow(2)
1270     IF NOT fast THEN putshape(inq(19),inq(21),shape$)
1280     x0:=inq(23); y0:=y; v:=0
1290     moveto(x0,y0)
1300     REPEAT
1310         m:=0; tmin:=1000; sv:=SIN(v); cv:=COS(v)
1320         FOR n:=1 TO ncy1 DO
1330             a(n):=(cy(n)-y0)*cv-(cx(n)-x0)*sv
1340             IF ABS(a(n))<=r(n) THEN
1350                 b(n):=SQR(r(n)^2-a(n)^2)
1360                 t:=cv*(cx(n)-x0)+sv*(cy(n)-y0)-b(n)
1370                 IF (t>=0) AND (t<tmin) THEN tmin:=t; m:=n
1380             ENDIF
1390         ENDFOR n
1400         n:=m; t:=tmin
1410         IF n>0 THEN
1420             x0:=t*cv; y0:=t*sv; i:=ATN(a(n)/b(n)); v1:=v+PI+2*i
1430             drawto(x0,y0)
1440             WHILE (v1<0) DO v1:=+2*PI
1450             WHILE (v1>2*PI) DO v1:=-2*PI
1460             v:=v1
1470             IF lyd THEN play_tone(400/r(n)^1.5,0.2*ABS(COS(i)))
1480         ENDIF
1490     UNTIL n=0
1500     drawto(x0+8*COS(v),y0+8*SIN(v))
1510 ENDPROC ray
1520
1530
1540 PROC selectwindow(n) // skift mellem graf og billede
1550     CASE n OF
1560         WHEN 1 // grafvindue
1570             viewport(0,0,25*xpixels-1,1/25*ypixels+1,ypixels)
1580             window(0,2*PI,ystart,yslut)
1590         WHEN 2 // billede
1600             viewport(0,25*xpixels,xpixels,1/25*ypixels+1,ypixels)
1610             ratio:=aspect*((inq(20)-inq(19))/(inq(22)-inq(21)))
1620             IF ratio>1 THEN
1630                 window(-2,2,ratio*(-2),ratio*2)
1640             ELSE
1650                 window(-2/ratio,2/ratio,-2,2)
1660             ENDIF
1670         WHEN 3 // textvindue
1680             textwindow(25,25,1,79)
1690     OTHERWISE
1700     ENDCASE
1710 ENDPROC selectwindow
1720
1730 PROC cylindre
1740     selectwindow(3)
1750     getnumber("Antal cylindre = ",ncyl)
1760     getnumber("Middel-radius = ",mr)
1770     getnumber("Spredning på radius = ",sr)
```

```
1780 ncyl:=minmax(0,ncyl,ncylmax)
1790 mr:=minmax(0.1,mr,2)
1800 sr:=minmax(0,sr,2)
1810 FOR n:=1 TO ncyl DO
1820   r(n):=minmax(0.1,SQR(3)*(RND+RND+RND+RND-2)*sr+mr,2)
1830   trials:=0
1840   REPEAT
1850     cxx:=(RND-0.5)*(4-2*r(n)); cyy:=(RND-0.5)*(4-2*r(n))
1860     ok:=TRUE; m:=1
1870     WHILE ok AND m<n DO
1880       ok:=(cxx-cx(m))^2+(cyy-cy(m))^2>(r(m)+r(n))^2
1890       m:=+1
1900     ENDWHILE
1910     trials:=+1
1920     UNTIL ok OR (trials>=1000)
1930     cx(n):=cxx; cy(n):=cyy
1940   ENDFOR n
1950   drawcylindre
1960 ENDPROC cylindre
1970
1980 PROC infoline
1990   selectwindow(3)
2000   PRINT AT 1,1: USING f$: ystart;
2010   PRINT AT 1,20: USING f$: y;
2020   PRINT AT 1,40: USING f$: yslut;
2030   IF lyd THEN
2040     inline$(4:6):="LYD"
2050   ELSE
2060     inline$(4:6):=" "
2070   ENDIF
2080   IF scanning THEN
2090     inline$(10:17):="SCAN"
2100   ELIF zooming THEN
2110     inline$(10:17):="ZOOM"
2120   ELSE
2130     inline$(10:17):=" "
2140   ENDIF
2150   PRINT AT 1,60: inline$;
2160 ENDPROC infoline
2170
2180 FUNC max(x,y)
2190   IF x<y THEN
2200     RETURN y
2210   ELSE
2220     RETURN x
2230   ENDIF
2240 ENDFUNC max
2250
2260 FUNC min(x,y)
2270   IF x<y THEN
2280     RETURN x
2290   ELSE
2300     RETURN y
2310   ENDIF
2320 ENDFUNC min
2330
2340 FUNC minmax(xmin,x,xmax)
2350   RETURN min(xmax,max(xmin,x))
2360 ENDFUNC minmax
2370
2380 PROC drawcylindre
2390   selectwindow(2)
2400   clearwindow
2410   FOR n:=1 TO ncyl DO
```



```
2420     circle(cx(n),cy(n),r(n))
2430     fill(cx(n),cy(n))
2440   ENDFOR n
2450   getshape(inq(19),inq(20),inq(21),inq(22),shape$)
2460 ENDPROC drawcylindre
2470
2480 PROC getnumber(prompt$,REF number)
2490   TRAP
2500     PRINT AT 1,1: ""0" "+prompt$,number;
2510     INPUT AT 1,LEN(prompt$)+1,20: "": number;
2520   HANDLER
2530     PRINT ""7"";
2540   RETRY
2550   ENDTRAP
2560 ENDPROC getnumber
2570 PROC swap(REF x,REF y) CLOSED
2580   temp:=x; x:=y; y:=temp
2590 ENDPROC swap
2600
2610 PROC reset_scan
2620   scanning:=FALSE
2630   ystep:=(yslut-ystart)/nstep
2640   y:=ystart; ynext:=ystart
2650   selectwindow(1)
2660   clearwindow
2670 ENDPROC reset_scan
2680 PROC diagram
2690   selectwindow(1)
2700   moveto(0,y)
2710   drawto(v,y)
2720 ENDPROC diagram
2730 PROC clearwindow
2740   clear
2750   moveto(inq(23),inq(25))
2760   drawto(inq(24),inq(25))
2770   drawto(inq(24),inq(26))
2780   drawto(inq(23),inq(26))
2790   drawto(inq(23),inq(25))
2800 ENDPROC clearwindow
2810 PROC zoom
2820
2830   PROC level(y)
2840     selectwindow(1)
2850     moveto(0,y)
2860     drawto(2*PI,y)
2870     selectwindow(2)
2880     moveto(inq(25),y)
2890     drawto(inq(25)+0.05,y)
2900   ENDPROC level
2910
2920   pencolor(3) // XOR-White
2930   bottom:=TRUE
2940   zooming:=TRUE
2950   infoline
2960   selectwindow(1)
2970   y:=ystart; yst:=ystart+ystep; ysl:=yslut-ystep
2980   level(yst)
2990   level(ysl)
3000   REPEAT
3010     c$:=KEY$
3020     level(y) // sluk
3030     CASE c$ OF
3040       WHEN ""9"", ""27"", ""13""
3050         level(y) // tænd
```

```
3060      IF bottom THEN
3070          yst:=y
3080          y:=ysl
3090      ELSE
3100          ysl:=y
3110          y:=yst
3120      ENDIF
3130      level(y) // sluk
3140      bottom:=NOT bottom
3150      WHEN " "
3160          fast:=FALSE
3170          pencolor(1)
3180          ray
3190          pencolor(3)
3200          infoline
3210      WHEN "1"
3220          lyd:=NOT lyd
3230          infoline
3240      WHEN "'0"H" // op
3250          y:=min(ymax,y+ystep)
3260      WHEN "'0"P" // ned
3270          y:=max(ymin,y-ystep)
3280      WHEN "'0"I" // PgUp
3290          y:=min(ymax,y+10*ystep)
3300      WHEN "'0"Q" // PgDn
3310          y:=max(ymin,y-10*ystep)
3320      WHEN "'0"G" // Home
3330          y:=yslut-ystep
3340      WHEN "'0"O" // End
3350          y:=ystart+ystep
3360      OTHERWISE
3370      ENDCASE
3380      level(y) // tand
3390      UNTIL c$="27"
3400      level(yst) // sluk
3410      level(ysl) // sluk
3420      IF ysl<yst THEN swap(yst,ysl)
3430      IF ysl=yst THEN
3440          ysl:=ystep
3450          yst:=-ystep
3460      ENDIF
3470      pencolor(1)
3480      ystart:=yst; yslut:=ysl
3490      zooming:=FALSE
3500      ENDPROC zoom
```

Appendix B :

Programliste, bold i grøft .

```
0010 // Bold på skråplaner 30-11-88 Heine Larsen
0020 USE graphics
0030 USE sound
0040 // graphicscreen(0) // kan forcere anden mode end default
0050 www // amstrad koordinater
0060 scrfil$="d:bold.scr" // angiv et drev, som er hurtigt.
0070 // helst et RAM-drev eller en harddisk.
0080 cirk:=FALSE
0090 singlestep:=FALSE
0100 slut:=FALSE
0110 alfa:=1.0863
0120 lyd:=TRUE
0130 delaycount:=80
0140 DIM xg(0:639)
0150 REPEAT
0160 CURSOR 1,1
0170 PRINT "BOLD I GRØFT: C:Clear A:Alfa P:Poincare-map
      S:simulering";
0180 c$:=INKEY$
0190 CURSOR 1,1
0200 PRINT "
0210 CASE c$ OF
0220 WHEN "a"
0230 CURSOR 2,1
0240 PRINT "alfa = ",alfa;
0250 INPUT ", ny værdi: alfa = ": alfa
0260 WHEN "c"
0270 cirk:=FALSE
0280 WHEN "s"
0290 simulering
0300 cirk:=FALSE
0310 WHEN "p"
0320 poincare_map
0330 WHEN CHR$(27)
0340 textscreen
0350 DELETE scrfil$
0360 END
0370 OTHERWISE
0380 ENDCASE
0390 UNTIL FALSE
0400
0410 PROC poincare_map
0420 IF cirk THEN
0430 loadscreen(scrfil$)
0440 ELSE
0450 clearscreen
0460 moveto(20,64)
0470 drawto(620,64)
0480 cx:=320
0490 cy:=64
0500 rx:=300
0510 ry:=300
0520 FOR t:=0 TO PI STEP PI/180 DO
      drawto(cx+rx*COS(t),cy+ry*SIN(t))
```

```
0530    savescreen(scrfil$) // hvis der opstår fejl her,
0540    // så check værdien af scrfil$, om det er et lovligt
        filnavn.
0550    // Scrfil$ defineres i starten af programmet.
0560    cirk:=TRUE
0570    ENDIF
0580    CURSOR 25,1
0590    INPUT "ksi = alfa*x0/ym (mellem -1 og 1) ": ksi;
0600    xm:=45000*ksi/alfa
0610    loadscreen(scrfil$)
0620    PRINT AT 3,1: "ksi = ",ksi
0630    sm:=SGN(xm)
0640    IF sm=0 THEN sm=-1
0650    eta:=SQR(1-ksi)
0660    ca:=1/SQR(1+alfa^2)
0670    sa:=alfa*ca
0680    vtm:=-300*sa*sm*eta
0690    vnm:=300*ca*eta
0700    vp:=ATN(ABS(alfa))
0710    am:=1-alfa^2
0720    ap:=1+alfa^2
0730    WHILE KEY$="" DO
0740        plot(320+vtm,64+vnm)
0750        vxm:=vtm-alfa*sm*vnm
0760        xmpl:=xm+2*vnm*vxm
0770        IF SGN(xmpl)=sm THEN
0780            xm:=xmpl
0790            vtm:=vtm-2*alfa*sm*vnm
0800        ELSE
0810            wm:=SQR(180000-2*vnm^2+am*(am*vnm^2-2*vtm^2+
                4*sm*alfa*vtm*vnm)/ap)
0820            xm:=xm+ca*vxm*(ca*(2*alfa*sm*vtm+am*vnm)+wm)
0830            vtm:=vtm-alfa*sm*vnm+sa*sm*wm
0840            vnm:=ca*wm
0850            sm:=-sm
0860        ENDIF
0870    ENDWHILE
0880    savescreen(scrfil$)
0890    ENDPROC poicare_map
0900
0910    FUNC min(x1,x2)
0920    IF x1>x2 THEN
0930        RETURN x2
0940    ELSE
0950        RETURN x1
0960    ENDIF
0970    ENDFUNC min
0980
0990    PROC simulering
1000    clearscreen
1010    vp:=ATN(ABS(alfa))
1020    www
1030    REPEAT
1040        CURSOR 1,1
1050        INPUT "max. højde, 0 < ym < 200, ym = ": ym
1060        CURSOR 1,1
1070        PRINT "
1080        CURSOR 1,1
1090        IF ym=0 THEN ym:=min(alfa*160,200)
1100        xp0:=min(ym/alfa,160)
1110        UNTIL (ym>=0) AND (ym<=200)
1120        moveto(320,0)
1130
```

```
1140 IF xp0<160 THEN xp0:=200/alfa
1150 IF xp0<160 THEN
1160   drawto(2*xp0+320,400)
1170 ELSE
1180   drawto(640,alfa*320)
1190 ENDIF
1200 moveto(320,0)
1210 IF xp0<160 THEN
1220   drawto(320-2*xp0,400)
1230 ELSE
1240   drawto(0,alfa*320)
1250 ENDIF
1260 fill(100,0)
1270 fill(400,0)
1280 IF ym<200 THEN
1290   moveto(0,2*ym)
1300   drawto(640,2*ym)
1310 ENDIF
1320 savescreen(scrfil$)
1330 CURSOR 1,1
1340 PRINT "startværdi for x, ";-xp0;" < x0 < ";xp0
1350 REPEAT
1360   INPUT " x0 = ": x00
1370 UNTIL (x00>-xp0) AND (x00<xp0)
1380 x0:=x00
1390 xp:=320+2*x0
1400 yp:=2*ym
1410 loadscreen(scrfil$)
1420 www
1430 moveto(xp,yp)
1440 draw(0,-2)
1450 WHILE KEY$="" DO NULL
1460 // lodret fald fra højden ym
1470 moveto(xp,yp)
1480
1490 ng:=0
1500 xg(0):=x0
1510 t1:=SQR(2*(ym-alfa*ABS(x0)))
1520 t1i:=INT(t1+1)
1530 FOR t:=1 TO t1i DO drawto(xp,yp-t*t)
1540 IF lyd THEN dunk(x0)
1550 th0:=2*vp*SGN(x0)+PI/2
1560 // beregn og tegn parabel givet x0 og th0
1570 singlestep:=FALSE
1580 REPEAT
1590   ng:=ng+1
1600   IF ng=639 THEN // plot kurve for xg(t)
1610     ng:=0
1620   ENDIF
1630   // beregn næste nedslag
1640   y0:=alfa*ABS(x0)
1650   sx0:=SGN(x0)
1660   ax0:=alfa*sx0
1670   v02:=2*(ym-y0)
1680   v0:=SQR(v02)
1690   ct:=COS(th0)
1700   st:=SIN(th0)
1710   vx:=-v0*ct
1720   vy:=-v0*st
1730   IF th0=PI/2 THEN
1740     t1:=2*v0
1750     x1:=x0
1760   ELSE
```

```
1770      b0:=st/ct
1780      c2t:=ct*ct
1790      delx:=2*(b0-ax0)*v02*c2t
1800      x1:=x0+delx
1810      IF SGN(x1)<>sx0 THEN
1820          dba:=b0+ax0
1830          delx:=dba*v02*c2t
1840          sqt:=SQR(1+4*y0/(v02*c2t*dba^2))
1850          x1:=x0+delx*(1+sqt)
1860          IF SGN(x1)=sx0 THEN
1870              x1:=x0+delx*(1-sqt)
1880          ENDIF
1890      ENDIF
1900      t1:=(x1-x0)/vx
1910  ENDIF
1920  xg(ng):=x0
1930  tli:=INT(t1+1)
1940  moveto(320+2*x0,2*y0)
1950  FOR t:=1 TO tli DO
1960      drawto(320+2*(x0+vx*t),2*(y0+vy*t)-t*t)
1970      delay
1980  ENDFOR t
1990  IF lyd THEN dunk(x1)
2000  // beregn sluthastighed og ny udfaldsvinkel
2010  vy:=vy-t1
2020  // dv bliver retningen i rad af (vx,vy)
2030  IF vx=0 THEN
2040      dv:=SGN(vy)*PI/2
2050  ELIF vy=0 THEN
2060      dv:=(1-SGN(vx))*PI/2
2070  ELSE
2080      dv:=ATN(vy/vx)
2090
2100      IF dv<0 THEN
2110          dv:=dv+(vy>0)*PI
2120      ELSE
2130          dv:=dv-(vx<0)*PI
2140      ENDIF
2150  ENDIF
2160  th0:=2*vp*SGN(x1)-dv
2170  x0:=x1
2180  IF singlestep THEN
2190      PRINT AT 25,1: th0," ",x0," ";
2200      a$:=INKEY$
2210  ELSE
2220      a$:=KEY$
2230  ENDIF
2240  CASE a$ OF
2250  WHEN "s"
2260      singlestep:=NOT singlestep
2270  WHEN "c"
2280      loadscreen(scrfil$)
2290  WHEN "l"
2300      lyd:=NOT lyd
2310  WHEN " "
2320      a$:=INKEY$
2330  WHEN "k"
2340      clearscreen
2350      FOR t:=ng+1 TO 639 DO plot(t-ng,(xg(t)+160)*1.25)
2360      FOR t:=0 TO ng DO plot(t+(639-ng),(xg(t)+160)*1.25)
2370      WHILE KEY$="" DO NULL
2380      loadscreen(scrfil$)
2390  WHEN "+"
2400      delaycount:=0.9*delaycount
```

```
2410     WHEN "-"
2420         delaycount:=1.1*delaycount
2430     OTHERWISE
2440     ENDCASE
2450     UNTIL a$=CHR$(27)
2460
2470
2480 ENDPROC simulering
2490
2500 PROC www
2510     window(0.639,0.399)
2520 ENDPROC www
2530
2540 PROC dunk(x)
2550     play_tone(5*ABS(x)+160,0.1)
2560 ENDPROC dunk
2570
2580 PROC delay
2590     FOR i:=0 TO delaycount DO NULL
2600 ENDPROC delay
```

- 1/78 "TANKER OM EN PRAKSIS" - et matematikprojekt. Projektrapport af: Anne Jensen, Lena Lindenskov, Marianne Kesselhahn og Nicolai Lomholt. Vejleder: Anders Madsen
- 2/78 "OPTIMERING" - Menneskets forøgede beherskelsesmuligheder af natur og samfund. Projektrapport af: Tom J. Andersen, Tommy R. Andersen, Gert Krenøe og Peter H. Lassen. Vejleder: Bernhelm Boss.
- 3/78 "OPCAVESAMLING", breddekursus i fysik. Af: Lasse Rasmussen, Aage Bonde Kræmmer og Jens Højgaard Jensen.
- 4/78 "TRE ESSAYS" - om matematikundervisning, matematiklærerruddannelsen og videnskabsrindalismen. Af: Mogens Niss. Nr. 4 er p.t. udgået.
- 5/78 "BIBLIOGRAFISK VEJLEDNING til studiet af DEN MODERNE FYSIKS HISTORIE". Af: Helge Kragh. Nr. 5 er p.t. udgået.
- 6/78 "NOGLE ARTIKLER OG DEBATINDLÆG OM - læreruddannelse og undervisning i fysik, og - de naturvidenskabelige fags situation efter studenteroprøret". Af: Karin Beyer, Jens Højgaard Jensen og Bent C. Jørgensen.
- 7/78 "MATEMATIKKENS FORHOLD TIL SAMFUNDSØKONOMIEN". Af: B.V. Gnedenko. Nr. 7 er udgået.
- 8/78 "DYNAMIK OG DIAGRAMMER". Introduktion til energy-bond-graph formalismen. Af: Peder Voetmann Christiansen.
- 9/78 "OM PRAKSIS' INDFLYDELSE PÅ MATEMATIKKENS UDVIKLING". - Motiver til Kepler's: "Nova Stereometria Doliorum Vinariorum". Projektrapport af: Lasse Rasmussen. Vejleder: Anders Madsen.
-
- 10/79 "THERMODYNAMIK I GYMNASIET". Projektrapport af: Jan Christensen og Jeanne Mortensen. Vejledere: Karin Beyer og Peder Voetmann Christiansen.
- 11/79 "STATISTISKE MATERIALER". Af: Jørgen Larsen.
- 12/79 "LINEÆRE DIFFERENTIALLIGNINGER OG DIFFERENTIALLIGNINGSSYSTEMER". Af: Mogens Brun Heefelt. Nr. 12 er udgået.
- 13/79 "CAVENDISH'S FORSØG I GYMNASIET". Projektrapport af: Gert Kreinøe. Vejleder: Albert Chr. Paulsen.
- 14/79 "BOOKS ABOUT MATHEMATICS: History, Philosophy, Education, Models, System Theory, and Works of". Af: Else Høyrup. Nr. 14 er p.t. udgået.
- 15/79 "STRUKTUREL STABILITET OG KATASTROFER i systemer i og udenfor termodynamisk ligevægt". Specialeopgave af: Leif S. Striegler. Vejleder: Peder Voetmann Christiansen.
- 16/79 "STATISTIK I KRÆFTFORSKNINGEN". Projektrapport af: Michael Olsen og Jørn Jensen. Vejleder: Jørgen Larsen.
- 17/79 "AT SPØRGE OG AT SVARE i fysikundervisningen". Af: Albert Christian Paulsen.
- 18/79 "MATHEMATICS AND THE REAL WORLD", Proceedings af an International Workshop, Roskilde University Centre, Denmark, 1978. Preprint. Af: Bernhelm Booss og Mogens Niss (eds.)
- 19/79 "GEOMETRI, SKOLE OG VIRKELIGHED". Projektrapport af: Tom J. Andersen, Tommy R. Andersen og Per H.H. Larsen. Vejleder: Mogens Niss.
- 20/79 "STATISTISKE MODELLER TIL BESTEMMELSE AF SIKRE DOSER FOR CARCINOGENE STOFFER". Projektrapport af: Michael Olsen og Jørn Jensen. Vejleder: Jørgen Larsen.
- 21/79 "KONTROL I GYMNASIET-FORMÅL OG KONSEKVENSER". Projektrapport af: Crilles Bacher, Per S. Jensen, Preben Jensen og Torben Nysteen.
- 22/79 "SEMIOTIK OG SYSTEMEGENSKABER (I)". I-port lineært response og støj i fysikken. Af: Peder Voetmann Christiansen.
- 23/79 "ON THE HISTORY OF EARLY WAVE MECHANICS - with special emphasis on the role of reality". Af: Helge Kragh.
-
- 24/80 "MATEMATIKOPFATTELSE HOS 2.G'ERE". a+b 1. En analyse. 2. Interviewmateriale. Projektrapport af: Jan Christensen og Knud Lindhardt Rasmussen. Vejleder: Mogens Niss.
- 25/80 "EKSAMENSOPGAVER", Dybdemodulet/fysik 1974-79.
- 26/80 "OM MATEMATISKE MODELLER". En projektrapport og to artikler. Af: Jens Højgaard Jensen m.fl.
- 27/80 "METHODODOLOGY AND PHILOSOPHY OF SCIENCE IN PAUL DIRAC'S PHYSICS". Af: Helge Kragh.
- 28/80 "DILETTANTISK RELAXATION - et forslag til en ny model bygget på væskernes viscoelastiske egenskaber". Projektrapport af: Gert Kreinøe. Vejleder: Niels Boye Olsen.
- 29/80 "ODIN - undervisningsmateriale til et kursus i differentiaalligningsmodeller". Projektrapport af: Tommy R. Andersen, Per H.H. Larsen og Peter H. Lassen. Vejleder: Mogens Brun Heefelt.
- 30/80 "FUSIONSENERGIEN - - - ATOMSAMFUNDETS ENDESTATION". Af: Oluf Danielsen. Nr. 30 er udgået.
- 31/80 "VIDENSKABSTEORETISKE PROBLEMER VED UNDERVISNINGSSYSTEMER BASERET PÅ MÅNGDELÆRE". Projektrapport af: Troels Lange og Jørgen Karrebæk. Vejleder: Stig Andur Pedersen. Nr. 31 er p.t. udgået.
- 32/80 "POLYMERE STOFFERS VISCOELASTISKE EGENSKABER - BELYST VED HJÆLP AF MEKANISKE IMPEDANSMÅLINGER - GER MOSSBAUEREFFEKT MÅLINGER". Projektrapport af: Crilles Bacher og Preben Jensen. Vejledere: Niels Boye Olsen og Peder Voetmann Christiansen.
- 33/80 "KONSTITUERING AF FAG INDEN FOR TEKNISK - NATURVIDENSKABELIGE UDDANNELSER. I-II". Af: Arne Jakobsen.
- 34/80 "ENVIRONMENTAL IMPACT OF WIND ENERGY UTILIZATION". ENERGY SERIES NO. 1. Af: Bent Sørensen. Nr. 34 er udgået.

35/80 "HISTORISKE STUDIER I DEN NYERE ATOMFYSIKS UDVIKLING".
Af: Helge Kragh.

36/80 "HVAD ER MENINGEN MED MATEMATIKUNDERVISNINGEN?".
Fire artikler.
Af: Mogens Niss.

37/80 "RENEWABLE ENERGY AND ENERGY STORAGE".
ENERGY SERIES NO. 2.
Af: Bent Sørensen.

38/81 "TIL EN HISTORIETBORI OM NATURERKENDELSE, TEKNOLOGI OG SAMFUND".
Projektrapport af: Erik Gade, Hans Hedal, Henrik Lau og Finn Physant.
Vejledere: Stig Andur Pedersen, Helge Kragh og Ib Thiersen.
Nr. 38 er p.t. udgået.

39/81 "TIL KRITIKKEN AF VÆKSTØKONOMIEN".
Af: Jens Højgaard Jensen.

40/81 "TELEKOMMUNIKATION I DANMARK - oplæg til en teknologivurdering".
Projektrapport af: Arne Jørgensen, Bruno Petersen og Jan Vedde.
Vejleder: Per Nørgaard.

41/81 "PLANNING AND POLICY CONSIDERATIONS RELATED TO THE INTRODUCTION OF RENEWABLE ENERGY SOURCES INTO ENERGY SUPPLY SYSTEMS".
ENERGY SERIES NO. 3.
Af: Bent Sørensen.

42/81 "VIDENSKAB TEORI SAMFUND - En introduktion til materialistiske videnskabsopfattelser".
Af: Helge Kragh og Stig Andur Pedersen.

43/81 1. "COMPARATIVE RISK ASSESSMENT OF TOTAL ENERGY SYSTEMS".
2. "ADVANTAGES AND DISADVANTAGES OF DECENTRALIZATION".
ENERGY SERIES NO. 4.
Af: Bent Sørensen.

44/81 "HISTORISKE UNDERSØGELSER AF DE EKSPERIMENTELLE FORUDSÆTNINGER FOR RUTHERFORDS ATOMMODEL".
Projektrapport af: Niels Thor Nielsen.
Vejleder: Bent C. Jørgensen.

45/82 Er aldrig udkommet.

46/82 "EKSEMPLARISK UNDERVISNING OG FYSISK ERKENDELSE-
1+1 ILLUSTRERET VED TO EKSEMPLER".
Projektrapport af: Torben O. Olsen, Lasse Rasmussen og Niels Dreyer Sørensen.
Vejleder: Bent C. Jørgensen.

47/82 "BARSEBÄCK OG DET VÆRST OFFICIELT-TÆNKELIGE UHELD".
ENERGY SERIES NO. 5.
Af: Bent Sørensen.

48/82 "EN UNDERSØGELSE AF MATEMATIKUNDERVISNINGEN PÅ ADGANGSKURSUS TIL KØBENHAVNS TEKNIKUM".
Projektrapport af: Lis Eilertzen, Jørgen Karrebæk, Troels Lange, Preben Nørregaard, Lissi Pedersen, Lauust Rishøj, Lill Røn og Isac Showiki.
Vejleder: Mogens Niss.

49/82 "ANALYSE AF MULTISPEKTRALE SATELLITBILLEDER".
Projektrapport af: Preben Nørregaard.
Vejledere: Jørgen Larsen og Rasmus Ole Rasmussen.

50/82 "HERSLEV - MULIGHEDER FOR VEDVARENDE ENERGI I EN LANDSBY".
ENERGY SERIES NO. 6.
Rapport af: Bent Christensen, Bent Hove Jensen, Dennis B. Møller, Bjarne Laursen, Bjarne Lillethorup og Jacob Mørch Pedersen.
Vejleder: Bent Sørensen.

51/82 "HVAD KAN DER Gøres FOR AT AFHJÆLPE PICERS BLOKERING OVERFOR MATEMATIK?".
Projektrapport af: Lis Eilertzen, Lissi Pedersen, Lill Røn og Susanne Stender.

52/82 "DESUSPENSION OF SPLITTING ELLIPTIC SYMBOLS".
Af: Bernhelm Booss og Krzysztof Wojciechowski.

53/82 "THE CONSTITUTION OF SUBJECTS IN ENGINEERING EDUCATION".
Af: Arne Jacobsen og Stig Andur Pedersen.

54/82 "FUTURES RESEARCH" - A Philosophical Analysis of Its Subject-Matter and Methods.
Af: Stig Andur Pedersen og Johannes Witt-Hansen.

55/82 "MATEMATISKE MODELLER" - Litteratur på Roskilde Universitetsbibliotek.
En biografi.
Af: Else Høytrup.

Vedr. tekst nr. 55/82 se også tekst nr. 62/83.

56/82 "EN - TO - MANGE" -
En undersøgelse af matematisk økologi.
Projektrapport af: Troels Lange.
Vejleder: Anders Madsen.

57/83 "ASPECT EKSPERIMENTET"-
Skjulte variable i kvantemekanikken?
Projektrapport af: Tom Juul Andersen.
Vejleder: Peder Voetmann Christiansen.
Nr. 57 er udgået.

58/83 "MATEMATISKE VANDRINGER" - Modelbetragtninger over spredning af dyr mellem småbiotoper i agerlandet.
Projektrapport af: Per Hammershøj Jensen og Lene Vagn Rasmussen.
Vejleder: Jørgen Larsen.

59/83 "THE METHODOLOGY OF ENERGY PLANNING".
ENERGY SERIES NO. 7.
Af: Bent Sørensen.

60/83 "MATEMATISK MODEKSPERTISE"- et eksempel.
Projektrapport af: Erik O. Gade, Jørgen Karrebæk og Preben Nørregaard.
Vejleder: Anders Madsen.

61/83 "FYSIKS IDEOLOGISKE FUNKTION, SOM ET EKSEMPEL PÅ EN NATURVIDENSKAB - HISTORISK SET".
Projektrapport af: Annette Post Nielsen.
Vejledere: Jens Høytrup, Jens Højgaard Jensen og Jørgen Vogelius.

62/83 "MATEMATISKE MODELLER" - Litteratur på Roskilde Universitetsbibliotek.
En biografi 2. rev. udgave.
Af: Else Høytrup.

63/83 "CREATING ENERGY FUTURES: A SHORT GUIDE TO ENERGY PLANNING".
ENERGY SERIES No. 8.
Af: David Crossley og Bent Sørensen.

64/83 "VON MATEMATIK UND KRIEG".
Af: Bernhelm Booss og Jens Høytrup.

65/83 "ANVENDT MATEMATIK - TEORI ELLER PRAKSIS".
Projektrapport af: Per Hedegård Andersen, Kirsten Habekost, Carsten Holst-Jensen, Annelise von Moos, Else Marie Pedersen og Erling Møller Pedersen.
Vejledere: Bernhelm Booss og Klaus Grünbaum.

66/83 "MATEMATISKE MODELLER FOR PERIODISK SELEKTION I ESCHERICHIA COLI".
Projektrapport af: Hanne Lisbet Andersen, Ole Richard Jensen og Klavs Frisdahl.
Vejledere: Jørgen Larsen og Anders Hede Madsen.

67/83 "ELEPSOIDE METODEN - EN NY METODE TIL LINEÆR PROGRAMMERING?".
Projektrapport af: Lone Billmann og Lars Boye.
Vejleder: Mogens Brun Heefelt.

68/83 "STOKASTISKE MODELLER I POPULATIONSGENETIK" - til kritikken af teoriladede modeller.
Projektrapport af: Lise Odgård Gade, Susanne Hansen, Michael Hviid og Frank Mølgård Olsen.
Vejleder: Jørgen Larsen.

- 69/83 "ELEVFORUDSÆTNINGER I FYSIK".
- en test i l.g med kommentarer.
Af: Albert C. Paulsen.
- 70/83 "INDLÆRINGS - OG FORMIDLINGSPROBLEMER I MATEMATIK PÅ VOKSENUNDERVISNINGSNIVEAU".
Projektrapport af: Hanne Lisbet Andersen, Torben J. Andraassen, Svend Åge Houmann, Helle Glerup Jensen, Keld Fl. Nielsen, Lene Vagn Rasmussen.
Vejleder: Klaus Grünbaum og Anders Hede Madsen.
- 71/83 "PIGER OG FYSIK"
- et problem og en udfordring for skolen?
Af: Karin Beyer, Sussanne Blegaa, Birthe Olsen, Jette Reich og Mette Vedelsby.
- 72/83 "VERDEN IFØLGE PEIRCE" - to metafysiske essays, om og af C.S. Peirce.
Af: Peder Voetmann Christiansen.
- 73/83 "'EN ENERGIANALYSE AF LANDBRUG"
- økologisk contra traditionelt.
ENERGY SERIES NO. 9
Specialeopgave i fysik af: Bent Hove Jensen.
Vejleder: Bent Sørensen.
-
- 74/84 "MINIATURISERING AF MIKROELEKTRONIK" - om videnskabeliggjort teknologi og nytten af at lære fysik.
Projektrapport af: Bodil Harder og Linda Szkotak Jensen.
Vejledere: Jens Højgaard Jensen og Bent C. Jørgensen.
- 75/84 "MATEMATIKUNDERVISNINGEN I FREMTIDENS GYMNASIUM"
- Case: Lineær programmering.
Projektrapport af: Morten Blomhøj, Klavs Frisdahl og Frank Mølgaard Olsen.
Vejledere: Mogens Brun Heefelt og Jens Bjørneboe.
- 76/84 "KERNEKRAFT I DANMARK?" - Et høringsvar indkaldt af miljøministeriet, med kritik af miljøstyrelsens rapporter af 15. marts 1984.
ENERGY SERIES No. 10
Af: Niels Boye Olsen og Bent Sørensen.
- 77/84 "POLITISKE INDEKS - FUP ELLER FAKTA?"
Opinionsundersøgelser belyst ved statistiske modeller.
Projektrapport af: Svend Åge Houmann, Keld Nielsen og Susanne Stender.
Vejledere: Jørgen Larsen og Jens Bjørneboe.
- 78/84 "JÆVNSTRØMSLEDNINGSEVNE OG GITTERSTRUKTUR I AMORFT GERMANIUM".
Specialrapport af: Hans Hedal, Frank C. Ludvigsen og Finn C. Physant.
Vejleder: Niels Boye Olsen.
- 79/84 "MATEMATIK OG ALMENDANNELSE".
Projektrapport af: Henrik Coster, Mikael Wennerberg Johansen, Povl Kattler, Birgitte Lydholm og Morten Overgaard Nielsen.
Vejleder: Bernhelm Booss.
- 80/84 "KURSUSMATERIALE TIL MATEMATIK B".
Af: Mogens Brun Heefelt.
- 81/84 "FREKVENSafhængig LEDNINGSEVNE I AMORFT GERMANIUM".
Specialrapport af: Jørgen Wind Petersen og Jan Christensen.
Vejleder: Niels Boye Olsen.
- 82/84 "MATEMATIK - OG FYSIKUNDERVISNINGEN I DET AUTOMATISEREDE SAMFUND".
Rapport fra et seminar afholdt i Hvidovre 25-27 april 1983.
Red.: Jens Højgaard Jensen, Bent C. Jørgensen og Mogens Niss.
- 83/84 "ON THE QUANTIFICATION OF SECURITY":
PEACE RESEARCH SERIES NO. 1
Af: Bent Sørensen
nr. 83 er p.t. udgået
- 84/84 "NOGLE ARTIKLER OM MATEMATIK, FYSIK OG ALMENDANNELSE".
Af: Jens Højgaard Jensen, Mogens Niss m. fl.
- 85/84 "CENTRIFUGALREGULATORER OG MATEMATIK".
Specialrapport af: Per Hedegård Andersen, Carsten Holst-Jensen, Else Marie Pedersen og Erling Møller Pedersen.
Vejleder: Stig Andur Pedersen.
- 86/84 "SECURITY IMPLICATIONS OF ALTERNATIVE DEFENSE OPTIONS FOR WESTERN EUROPE".
PEACE RESEARCH SERIES NO. 2
Af: Bent Sørensen.
- 87/84 "A SIMPLE MODEL OF AC HOPPING CONDUCTIVITY IN DISORDERED SOLIDS".
Af: Jeppe C. Dyre.
- 88/84 "RISE, FALL AND RESURRECTION OF INFINITESIMALS".
Af: Detlef Laugwitz.
- 89/84 "FJERNVARMEOPTIMERING".
Af: Bjarne Lillethorup og Jacob Mørch Pedersen.
- 90/84 "ENERGI I L.G. - EN TEORI FOR TILRETTELÆGGELSE".
Af: Albert Chr. Paulsen.
-
- 91/85 "KVANTETEORI FOR GYMNASIET".
1. Lærervejledning
Projektrapport af: Biger Lundgren, Henning Sten Hansen og John Johansson.
Vejleder: Torsten Meyer.
- 92/85 "KVANTETEORI FOR GYMNASIET".
2. Materiale
Projektrapport af: Biger Lundgren, Henning Sten Hansen og John Johansson.
Vejleder: Torsten Meyer.
- 93/85 "THE SEMIOTICS OF QUANTUM - NON - LOCALITY".
Af: Peder Voetmann Christiansen.
- 94/85 "TREENIGHEDEN BOURBAKI - generalen, matematikeren og ånden".
Projektrapport af: Morten Blomhøj, Klavs Frisdahl og Frank M. Olsen.
Vejleder: Mogens Niss.
- 95/85 "AN ALTERNATIV DEFENSE PLAN FOR WESTERN EUROPE".
PEACE RESEARCH SERIES NO. 3
Af: Bent Sørensen
- 96/85 "ASPEKTER VED KRAFTVARMEFORSYNING".
Af: Bjarne Lillethorup.
Vejleder: Bent Sørensen.
- 97/85 "ON THE PHYSICS OF A.C. HOPPING CONDUCTIVITY".
Af: Jeppe C. Dyre.
- 98/85 "VALGMULIGHEDER I INFORMATIONSALEDEREN".
Af: Bent Sørensen.
- 99/85 "Der er langt fra Q til R".
Projektrapport af: Niels Jørgensen og Mikael Klintorp.
Vejleder: Stig Andur Pedersen.
- 100/85 "TALSYSTEMETS OPBYGNING".
Af: Mogens Niss.
- 101/85 "EXTENDED MOMENTUM THEORY FOR WINDMILLS IN PERTURBATIVE FORM".
Af: Ganesh Sengupta.
- 102/85 OPSTILLING OG ANALYSE AF MATEMATISKE MODELLER, BELYST VED MODELLER OVER KØRS FODEROPTAGELSE OG - OMSETNING".
Projektrapport af: Lis Lilholtzen, Kirsten Habekost, Lill Røn og Susanne Stender.
Vejleder: Klaus Grünbaum.

- 103/85 "ØDSLE KOLDKRIGERE OG VIDENSKABENS LYSE IDEER".
Projekt rapport af: Niels Ole Dam og Kurt Jensen.
Vejleder: Bent Sørensen.
- 104/85 "ANALOGREGNEMASKINEN OG LORENZLIGNINGER".
Af: Jens Jäger.
- 105/85 "THE FREQUENCY DEPENDENCE OF THE SPECIFIC HEAT OF THE GLASS TRANSITION".
Af: Tage Christensen.
- "A SIMPLE MODEL OF AC HOPPING CONDUCTIVITY".
Af: Jeppe C. Dyre.
Contributions to the Third International Conference on the Structure of Non - Crystalline Materials held in Grenoble July 1985.
- 106/85 "QUANTUM THEORY OF EXTENDED PARTICLES".
Af: Bent Sørensen.
- 107/85 "EN MYG GØR INGEN EPIDEMI".
- flodblindhed som eksempel på matematisk modellering af et epidemiologisk problem.
Projekt rapport af: Per Hedegård Andersen, Lars Boye, Carsten Holst Jensen, Else Marie Pedersen og Erling Møller Pedersen.
Vejleder: Jesper Larsen.
- 108/85 "APPLICATIONS AND MODELLING IN THE MATHEMATICS CURRICULUM" - state and trends -
Af: Mogens Niss.
- 109/85 "COX I STUDIETIDEN" - Cox's regressionsmodel anvendt på studenteroplysninger fra RUC.
Projekt rapport af: Mikael Wennerberg Johansen, Poul Katler og Torben J. Andreasen.
Vejleder: Jørgen Larsen.
- 110/85 "PLANNING FOR SECURITY".
Af: Bent Sørensen
- 111/85 "JORDEN RUNDT PÅ FLADE KORT".
Projekt rapport af: Birgit Andresen, Beatriz Quinones og Jimmy Staal.
Vejleder: Mogens Niss.
- 112/85 "VIDENSKABELIGGØRELSE AF DANSK TEKNOLOGISK INNOVATION FREM TIL 1950 - BELYST VED EKSEMPLER".
Projekt rapport af: Erik Odgaard Gade, Hans Hedal, Frank C. Ludvigsen, Annette Post Nielsen og Finn Physant.
Vejleder: Claus Bryld og Bent C. Jørgensen.
- 113/85 "DESUSPENSION OF SPLITTING ELLIPTIC SYMBOLS II".
Af: Bernhard Booss og Krzysztof Wojciechowski.
- 114/85 "ANVENDELSE AF GRAFISKE METODER TIL ANALYSE AF KONFIGURATIONSTABELLER".
Projekt rapport af: Lone Billmann, Ole R. Jensen og Arne-Lise von Moos.
Vejleder: Jørgen Larsen.
- 115/85 "MATEMATIKKENS UDVIKLING OP TIL RENESSANCEN".
Af: Mogens Niss.
- 116/85 "A PHENOMENOLOGICAL MODEL FOR THE MEYER-NELDEL RULE".
Af: Jeppe C. Dyre.
- 117/85 "KRAFT & FJERNVARMEOPTIMERING".
Af: Jacob Mørch Pedersen.
Vejleder: Bent Sørensen
- 118/85 "TILFÆLDIGHEDEN OG NØDVENDIGHEDEN IFØLGE PEIRCE OG FYSIKKEN".
Af: Peder Voetmann Christiansen
- 119/86 "DET ER GANSKE VIST - - EUKLIDS FEMTE POSTULAT KUNNE NOK SKABE RØRE I ANDEDAMMEN".
Af: Ibén Maj Christiansen
Vejleder: Mogens Niss.
- 120/86 "ET ANTAL STATISTISKE STANDARDMODELLER".
Af: Jørgen Larsen
- 121/86 "SIMULATION I KONTINUERT TID".
Af: Peder Voetmann Christiansen.
- 122/86 "ON THE MECHANISM OF GLASS IONIC CONDUCTIVITY".
Af: Jeppe C. Dyre.
- 123/86 "GYMNASTIEFYSIKKEN OG DEN STORE VERDEN".
Fysiklærerforeningen, IMFUA, RUC.
- 124/86 "OPGAVESAMLING I MATEMATIK".
Samtlige opgaver stillet i tiden 1974-jan. 1986.
- 125/86 "UVBY, 8 - systemet - en effektiv fotometrisk spektral-klassifikation af B-, A- og F-stjerner".
Projekt rapport af: Birger Lundgren.
- 126/86 "OM UDVIKLINGEN AF DEN SPECIELLE RELATIVITETSTEORI".
Projekt rapport af: Lise Odgaard & Linda Szkotak Jensen
Vejledere: Karin Beyer & Stig Andur Pedersen.
- 127/86 "GALOIS' BIDRAG TIL UDVIKLINGEN AF DEN ABSTRAKTE ALGEBRA".
Projekt rapport af: Pernille Sand, Heine Larsen & Lars Frandsen.
Vejleder: Mogens Niss.
- 128/86 "SMÅKRYB" - en ikke-standard analyse.
Projekt rapport af: Niels Jørgensen & Mikael Klintorp.
Vejleder: Jeppe Dyre.
- 129/86 "PHYSICS IN SOCIETY"
Lecture Notes 1983 (1986)
Af: Bent Sørensen
- 130/86 "Studies in Wind Power"
Af: Bent Sørensen
- 131/86 "FYSIK OG SAMFUND" - Et integreret fysik/historie-projekt om naturanskuelsens historiske udvikling og dens samfundsmæssige betingethed.
Projekt rapport af: Jakob Heckscher, Søren Brønd, Andy Wierød.
Vejledere: Jens Højrup, Jørgen Vogelius, Jens Højgaard Jensen.
- 132/86 "FYSIK OG DANNEELSE"
Projekt rapport af: Søren Brønd, Andy Wierød.
Vejledere: Karin Beyer, Jørgen Vogelius.
- 133/86 "CHERNOBYL ACCIDENT: ASSESSING THE DATA. ENERGY SERIES NO. 15."
Af: Bent Sørensen.
-
- 134/87 "THE D.C. AND THE A.C. ELECTRICAL TRANSPORT IN AsSeTe SYSTEM"
Authors: M.B.El-Den, N.B.Olsen, Ib Høst Pedersen, Petr Visčor
- 135/87 "INTUITIONISTISK MATEMATIKS METODER OG ERKENDELSESTEORETISKE FORUDSÆTNINGER"
MATEMATIKSPECIALE: Claus Larsen
Vejledere: Anton Jensen og Stig Andur Pedersen
- 136/87 "Mystisk og naturlig filosofi: En skitse af kristendommens første og andet møde med græsk filosofi"
Projekt rapport af Frank Colding Ludvigsen
Vejledere: Historie: Ib Thiersen
Fysik: Jens Højgaard Jensen
- 137/87 "HOPMODELLER FOR ELEKTRISK LEDNING I UORDNEDE FASTE STOFFER" - Resume af licentiatafhandling
Af: Jeppe Dyre
Vejledere: Niels Boye Olsen og Peder Voetmann Christiansen.

- 138/87 "JOSEPHSON EFFECT AND CIRCLE MAP."
Paper presented at The International Workshop on Teaching Nonlinear Phenomena at Universities and Schools, "Chaos in Education". Balaton, Hungary, 26 April-2 May 1987.
By: Peder Voetmann Christiansen
- 139/87 "Machbarkeit nichtbeherrschbarer Technik durch Fortschritte in der Erkennbarkeit der Natur"
Af: Bernhelm Booss-Bavnbek
Martin Bohle-Carbonell
- 140/87 "ON THE TOPOLOGY OF SPACES OF HOLOMORPHIC MAPS"
By: Jens Gravesen
- 141/87 "RADIOMETERS UDVIKLING AF BLODGASAPPARATUR - ET TEKNOLOGIHISTORISK PROJEKT"
Projektrapport af Finn C. Physant
Vejleder: Ib Thiersen
- 142/87 "The Calderón Projektor for Operators With Splitting Elliptic Symbols"
by: Bernhelm Booss-Bavnbek og
Krzysztof P. Wojciechowski
- 143/87 "Kursusmateriale til Matematik på NAT-BAS"
af: Mogens Brun Heefelt
- 144/87 "Context and Non-Locality - A Peircan Approach
Paper presented at the Symposium on the Foundations of Modern Physics The Copenhagen Interpretation 60 Years after the Como Lecture. Joensuu, Finland, 6 - 8 august 1987.
By: Peder Voetmann Christiansen
- 145/87 "AIMS AND SCOPE OF APPLICATIONS AND MODELLING IN MATHEMATICS CURRICULA"
Manuscript of a plenary lecture delivered at ICMTA 3, Kassel, FRG 8.-11.9.1987
By: Mogens Niss
- 146/87 "BESTEMMELSE AF BULKRESISTIVITETEN I SILICIUM"
- en ny frekvensbaseret målemetode.
Fysikspeciale af Jan Vedde
Vejledere: Niels Boye Olsen & Petr Višćor
- 147/87 "Rapport om BIS på NAT-BAS"
redigeret af: Mogens Brun Heefelt
- 148/87 "Naturvidenskabsundervisning med Samfundsperspektiv"
af: Peter Colding-Jørgensen DLH
Albert Chr. Paulsen
- 149/87 "In-Situ Measurements of the density of amorphous germanium prepared in ultra high vacuum"
by: Petr Višćor
- 150/87 "Structure and the Existence of the first sharp diffraction peak in amorphous germanium prepared in UHV and measured in-situ"
by: Petr Višćor
- 151/87 "DYNAMISK PROGRAMMERING"
Matematikprojekt af:
Birgit Andresen, Keld Nielsen og Jimmy Staal
Vejleder: Mogens Niss
- 152/87 "PSEUDO-DIFFERENTIAL PROJECTIONS AND THE TOPOLOGY OF CERTAIN SPACES OF ELLIPTIC BOUNDARY VALUE PROBLEMS"
by: Bernhelm Booss-Bavnbek
Krzysztof P. Wojciechowski
- 153/88 "HALVLEDERTEKNOLOGIENS UDVIKLING MELLEM MILITÆRE OG CIVILE KRÆFTER"
Et eksempel på humanistisk teknologihistorie
Historiespeciale
Af: Hans Hedal
Vejleder: Ib Thiersen
- 154/88 "MASTER EQUATION APPROACH TO VISCOUS LIQUIDS AND THE GLASS TRANSITION"
By: Jeppe Dyre
- 155/88 "A NOTE ON THE ACTION OF THE POISSON SOLUTION OPERATOR TO THE DIRICHLET PROBLEM FOR A FORMALLY SELFADJOINT DIFFERENTIAL OPERATOR"
by: Michael Pedersen
- 156/88 "THE RANDOM FREE ENERGY BARRIER MODEL FOR AC CONDUCTION IN DISORDERED SOLIDS"
by: Jeppe C. Dyre
- 157/88 "STABILIZATION OF PARTIAL DIFFERENTIAL EQUATIONS BY FINITE DIMENSIONAL BOUNDARY FEEDBACK CONTROL: A pseudo-differential approach."
by: Michael Pedersen
- 158/88 "UNIFIED FORMALISM FOR EXCESS CURRENT NOISE IN RANDOM WALK MODELS"
by: Jeppe Dyre
- 159/88 "STUDIES IN SOLAR ENERGY"
by: Bent Sørensen
- 160/88 "LOOP GROUPS AND INSTANTONS IN DIMENSION TWO"
by: Jens Gravesen
- 161/88 "PSEUDO-DIFFERENTIAL PERTURBATIONS AND STABILIZATION OF DISTRIBUTED PARAMETER SYSTEMS: Dirichlet feedback control problems"
by: Michael Pedersen
- 162/88 "PIGER & FYSIK - OG MEGET MERE"
AF: Karin Beyer, Sussanne Blegaa, Birthe Olsen, Jette Reich, Mette Vedelsby
- 163/88 "EN MATEMATISK MODEL TIL BESTEMMELSE AF PERMEABILITETEN FOR BLOD-NETHINDE-BARRIEREN"
Af: Finn Langberg, Michael Jarden, Lars Frellesen
Vejleder: Jesper Larsen
- 164/88 "Vurdering af matematisk teknologi
Technology Assessment
Technikfolgenabschätzung"
Af: Bernhelm Booss-Bavnbek, Glen Pate med
Martin Bohle-Carbonell og Jens Højgaard Jensen
- 165/88 "COMPLEX STRUCTURES IN THE NASH-MOSER CATEGORY"
by: Jens Gravesen

166/88 "Grundbegreber i Sandsynligheds-
regningen"

Af: Jørgen Larsen

167a/88 "BASISSTATISTIK 1. Diskrete modeller"

Af: Jørgen Larsen

167b/88 "BASISSTATISTIK 2. Kontinuerte
modeller"

Af: Jørgen Larsen

168/88 "OVERFLADEN AF PLANETEN MARS"

Laboratorie-simulering og MARS-analoger
undersøgt ved Mössbauerspektroskopi.

Fysikspeciale af:

Birger Lundgren

Vejleder: Jens Martin Knudsen
Fys.Lab./HCØ

169/88 "CHARLES S. PEIRCE: MURSTEN OG MØRTEL
TIL EN METAFYSIK."

Fem artikler fra tidsskriftet "The Monist"
1891-93.

Introduktion og oversættelse:

Peder Voetmann Christéansen

170/88 "OPGAVESAMLING I MATEMATIK"

Samtlige opgaver stillet i tiden
1974 - juni 1988

171/88 "The Dirac Equation with Light-Cone Data"

af: Johnny Tom Ottesen